SURVIVAL KIT

Security Architect

Role, expectations and journey within the Digital Governance Framework (DGF): from the contract to the committee, from reading the file to the risk matrix.

Jeremy Canale

AWS / Azure Security Architect · September 2026

Where does the security architect belong?

Hard parts of the job

Expectations

Practical advice!

AGENDA

Table of contents

HOW TO USE

How to read this document

Executive

10 minutes

- Finding: a real DGF is still rare
- DGF gates
- Business as risk owner
- Four statuses of a file
- Ten pitfalls
- Setting up a DGF in 90 days

Project manager

30 minutes

- Three paths and checklist per gate
- Access the contract from day one
- Initial assessment form
- Complete checklist
- Matrix and risk card
- Case studies

Security architect

Everything, in order

- CIA+TN compass and its sub-criteria
- Thirty-six technical categories
- IAM, Azure and AWS: permissions
- AI project security
- Convincing and steering
- Patterns, pitfalls, toolbox, glossary

INTRODUCTION

Who am I, and why this kit?

Jeremy Canale

AWS / Azure Security Architect

More than 10 years of experience

contact@jeremycanale.com +33 7 87 77 55 92

Availability: France, Belgium, Switzerland

COUNTRIES

France · Belgium · Switzerland · Middle East

INDUSTRIES · LARGE ENTERPRISES ONLY

Finance · Banking · Defense · Insurance · Media · Retail · Pharmaceutical · ...

Why this survival kit?

Companies do not always understand where the security architect belongs: too early, too late, or outside the process. This kit gives concrete reference points to be in the right place at the right time, especially within the DGF.

10+
years of experience

4
geographic areas

100 %
large enterprises

INTRODUCTION

My journey

One of the first in Europe!

Before 2007 — Hacker

In my spare time, before making it a job

2011 — AWS

First steps on AWS; five certifications between 2016 and 2019

2013 — Cloud pioneer

One of the first cloud consultants in Europe

2020 — Azure

Azure Security Engineer Associate: security on both clouds

2023 — Security & governance

Cybersecurity Architect Expert, CISM, CRISC, Prisma Cloud

2026 — Artificial intelligence

AI Architect: Azure AI, NVIDIA, Anthropic

One of the first cloud consultants in Europe (2013)

Since then: AWS and Azure, security, governance… and now AI.

26
certifications and trainings

8
certification bodies

2
clouds: AWS and Azure

INTRODUCTION

My certifications

Microsoft

8 certifications

- Cybersecurity Architect Expert (2023)
- DevOps Engineer Expert (2023)
- Azure Security Engineer Associate (2020)
- Azure Developer Associate (2023)
- Security Operations Analyst Associate (2023)
- Azure AI Apps & Agents Developer Associate (2026)
- SC-900 · Security, Compliance & Identity (2023)
- AZ-900 · Azure Fundamentals (2023)

Amazon Web Services

5 certifications

- Security – Specialty (2019)
- SysOps Administrator – Associate (2018)
- Developer – Associate (2017)
- Solutions Architect – Associate (2016)
- Cloud Practitioner (2018)

Anthropic

6 trainings

- Building with the Claude API (2026)
- Claude Code in Action (2026)
- AI Fluency: Framework & Foundations (2026)
- Claude 101 · Claude Code 101 (2026)
- Introduction to Claude Cowork (2026)

ISACA

2 certifications

- CISM – Certified Information Security Manager (2023)
- CRISC – Certified in Risk and Information Systems Control (2023)

NVIDIA

2 certifications

- Certified Professional: Agentic AI (2026)
- Certified Associate: Generative AI LLMs (2026)

And also

3 certifications

- Palo Alto – Prisma Cloud Security Engineer (2023)
- Cisco – CCNA (2007)
- Scrum.org – Professional Scrum Master I (2019)

01

The security architect role

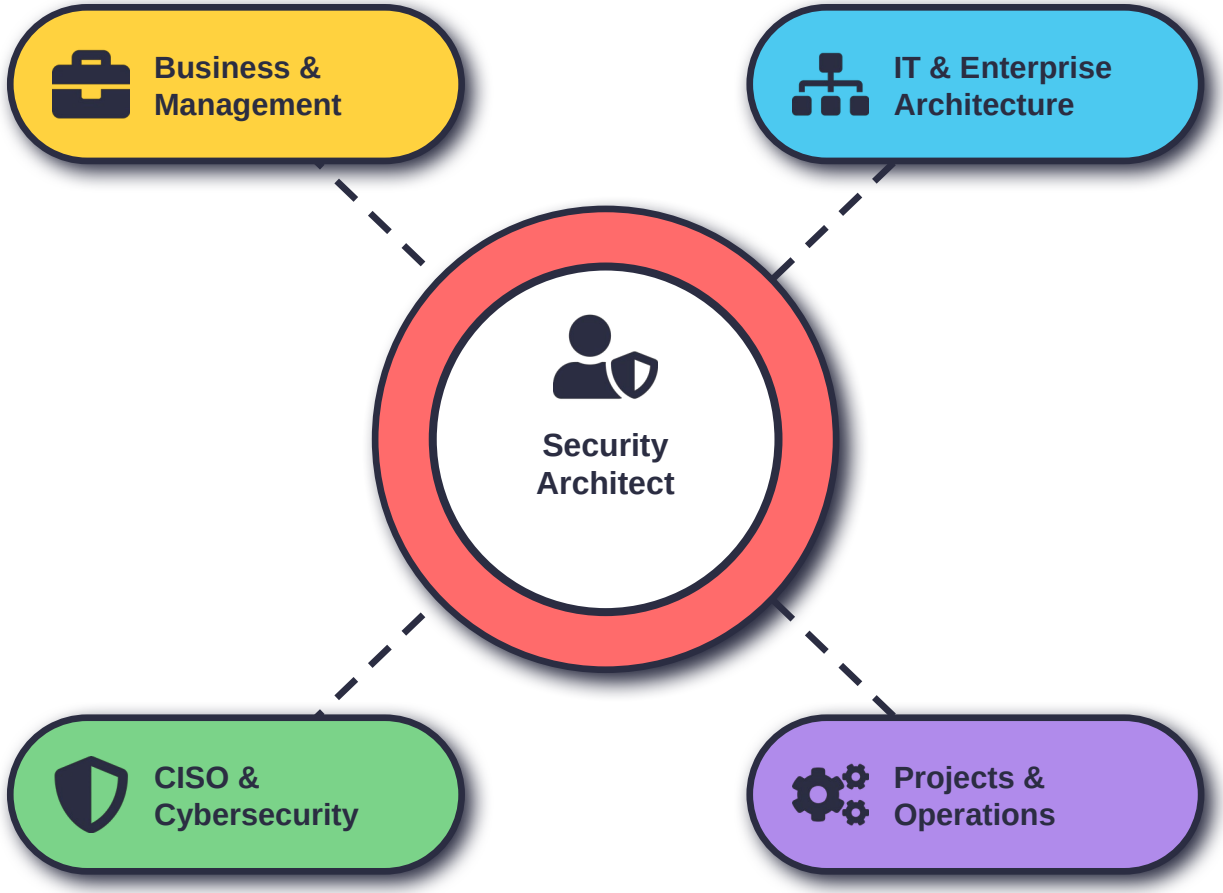
Who they are, what they do, and what is expected of them

PART 1 · THE ROLE

What does the security architect do?

The link between the business, the information system and cybersecurity.

They design secure architectures from the start, translate policies and risks into concrete requirements, and constantly arbitrate between protection level, feasibility and cost.



WHAT THEY ARE NOT

A firefighter — they act upstream, not only in reaction

A blocker — they look for the secure path rather than the "no"

A pure technician — they talk business risk as much as protocols

PART 1 · THE ROLE

Six main missions

Design

Define secure architectures: network, cloud, applications, identities, data.

Assess

Analyze risks and threats on projects, platforms and vendors.

Require

Translate security policies into verifiable technical requirements.

Control

Review the gates, review architecture files, track deviations.

Advise

Inform the decisions of the business, the IT department and the CISO with a reasoned opinion.

Anticipate

Keep watch on technology and regulation: NIS2, GDPR, DORA, new threats.

PART 1 · EXPECTATIONS

What is expected of them

Knowledge

Know

- Frameworks: ISO 27001, NIST CSF, CIS, OWASP
- Cloud, hybrid and network architectures
- Identity management and the Zero Trust model
- Cryptography, PKI, secrets management
- Regulatory and contractual framework

Know-how

Skills

- Model threats (STRIDE, EBIOS RM)
- Write a security architecture file
- Define reusable patterns and standards
- Run a design review
- Rate a risk and its treatment plan

Attitude

Posture

- Pragmatism: aim for acceptable risk, not zero risk
- Pedagogy: make security understandable
- Influence without hierarchical authority
- Know how to say no… and propose an alternative
- Sense of deadlines and project constraints

PART 1 · DELIVERABLES

Deliverables along a project

1 Scoping

- Security requirements of the need
- Data classification
- Initial risk analysis

2 Design

- Security architecture file
- Threat model
- Patterns and standards selected

3 Build

- Detailed design review
- Test requirements: SAST, DAST, penetration tests
- Deviation tracking

4 Go-live

- Opinion at the gate: green, yellow or red light
- Formalized exceptions
- Remediation plan

5 Operations

- Periodic architecture review
- Exception tracking
- Mapping update

PART 1 · THE KIT

Survival kit: 8 essentials

Mapping

Applications, flows, data, identities: you only protect what you know.

Requirements repository

Policies, standards, secure patterns and ready-to-use checklists.

Risk grid

Impact × probability, on a scale shared with the CISO and the business.

Data classification

Public, internal, confidential, secret: the level dictates the measures.

Network of allies

CISO, IT architects, procurement, legal, DPO, operations.

Exception register

Every accepted risk is recorded, signed, dated… and reviewed on time.

Three questions

Which data? Who accesses it? Where does it travel?

Pragmatism

Aim for acceptable risk, not zero risk: secure without blocking.

02

DGF

Digital Governance Framework: gates and three paths

PART 2 · THE DGF

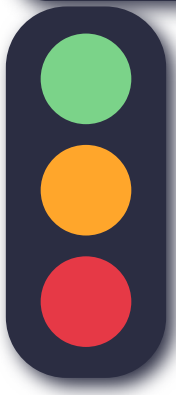
What is the DGF?

The Digital Governance Framework governs every significant change to the information system.

DGF · Digital Governance Framework

It applies whenever a new platform is integrated, a project is launched, or a merger or acquisition brings in an information system to absorb. The principle: a sequence of gates, each held by a body that gives an opinion before letting the project through.

AT EACH GATE, AN OPINION



Green light	Favorable opinion: on to the next gate
Yellow light	Favorable with conditions: dated action plan
Red light	Unfavorable opinion: back to design or stop

FOUR OBJECTIVES

Control risks

Identify and treat risks before commitment, not after.

Guarantee IT consistency

Avoid silos, duplicates and orphan technologies.

Ensure compliance

Regulatory, contractual and internal, with evidence.

Secure the investment

Decide on documented facts, not on promises.

PART 2 · FINDING

Finding: a real DGF is still rare

FIELD FINDING

30 %

of companies have a real DGF in place, with a dedicated application to track a project's life cycle.

Personal finding, based on the organizations I have worked with: not a formal study.

3 companies out of 10

With a real DGF

- Project life cycle visible end to end
- Gates, opinions and exceptions recorded and dated
- Indicators: lead times, blockers, accepted risks

Without a dedicated application

- Gates driven by emails and spreadsheets
- Decisions hard to find again
- Security consulted case by case, often late

PART 2 · THE GATES

DGF gates

You are here!

IT Gate

Who: IT department, enterprise architecture

Checks

- Strategic alignment and standards
- Ability to operate and to fund
- Impact on the existing IT

Security & Architecture Gate

Who: CISO, security architect

Checks

- Compliance with security policies
- Risk analysis
- Target architecture: flows, identities, data, hosting

Tech Readiness Gate

Who: Operations, production

Checks

- Environments and tests
- Monitoring and backups
- Recovery and operations plan

Governance Gate

Who: Governance committee

Decides

- Prioritization and budget
- Acceptance of residual risks
- Green light to launch

Sign-off

Who: Sponsors, management

Formalizes

- Signed commitment of the owners
- Scope, schedule, responsibilities
- Official launch

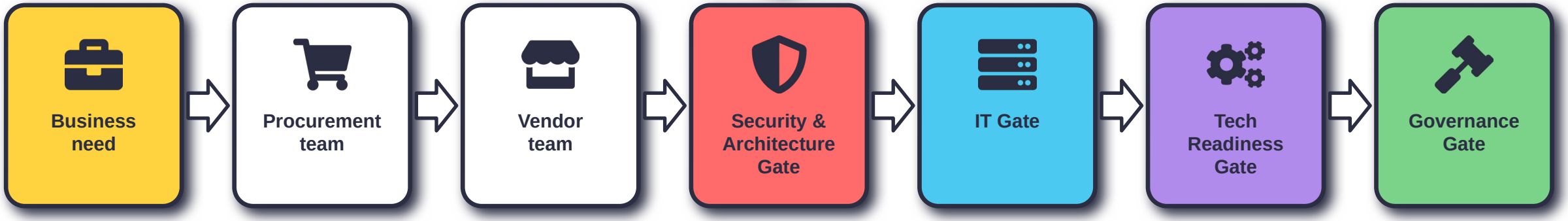
Original terminology: Gate IT · Gate Security & Architecture · Gate Tech Readiness · Gate Governance · Sign-off.

PART 2 · PATH 1

Path 1: new platform

SAP, CRM, SaaS platform, collaboration tool…

You are here!



Trigger

A business need the existing IT does not cover: ERP, CRM, SaaS platform, collaboration tool…

Specificity

Security qualifies the solution and the vendor before the IT department commits: here it comes first.

Watch point

Obtain the security clauses, certifications and reversibility plan from the vendor as early as the procurement phase.

PART 2 · PATH 2

Path 2: merger and acquisition

External growth, integration of an inherited IT

Trigger

An external growth operation: the acquired entity's IT must be integrated, isolated or decommissioned.

Specificity

The IT department first frames the technical scope; security then assesses the integration risks (acquisition audit).

Watch point

Do not interconnect the networks before the audit: inherited security debt, orphan identities, undeclared incidents.

PART 2 · PATH 3

Path 3: new project

Standard path for internal projects

Trigger

Any internal project that creates or significantly changes an application, a flow or a service.

Specificity

The only path closed by a formal sign-off: the sponsors commit to the scope and the residual risks.

Watch point

Involve security from the scoping phase: arrive at the gate with a file already reviewed, not with questions.

PART 2 · SUMMARY

Three paths, one constant

New platform

Path 1

Business need → Procurement → Vendor → Security & Architecture → IT → Tech readiness → Governance

Merger & acquisition

Path 2

Merger & acquisition → IT → Security & Architecture → Tech readiness → Governance

New project

Path 3

New project → IT → Security & Architecture → Tech readiness → Governance → Sign-off

What does not change

- The Security & Architecture gate is present in all three paths
- Tech Readiness then Governance always close the sequence
- Each gate gives an opinion: green, yellow or red light

What changes

- Trigger: business need, external growth or internal project
- IT / Security order: security comes first for a vendor platform
- A formal sign-off only exists for a new project

PART 2 · THE KEY MOMENT

At the Security & Architecture gate

What they expect as input

- Description of the need and the scope
- Target architecture diagram and flows
- Classification of the data processed
- Identity and access model
- Hosting and data location
- Vendor security clauses and evidence

What they check

- Compliance with policies and standards
- Exposure surface and segmentation
- Encryption in transit and at rest
- Logging, monitoring, detection
- Continuity, backups, recovery
- Regulatory compliance and reversibility

What they deliver

Green light

Compliant: on to the next gate.

Yellow light

Dated action plan, checked at the next gate.

Red light

Unacceptable risk: back to design, with a proposed alternative.

PART 2 · THE GATES

What you need before each gate

IT Gate

- [ ] Architecture file with a real diagram
- [ ] Alignment with the IT department's standards and catalog
- [ ] Costs and capacities estimated
- [ ] IT constraints integrated: DMZ, DNS, PKI, accounts, tools
- [ ] Impact on the existing IT described

Security & Architecture Gate

- [ ] Data classification and location
- [ ] Flow matrix
- [ ] Identity and access model: the seven IAM points
- [ ] Risk analysis and signed risk cards
- [ ] Contract and security annex read
- [ ] Penetration test report if the service is exposed

Tech Readiness Gate

- [ ] Environments ready, runbooks written
- [ ] Monitoring in place, logs connected to the SIEM
- [ ] Backups and restoration tested
- [ ] Handover to operations signed, on-call defined
- [ ] Conditions from previous gates closed

Governance Gate

- [ ] One-page summary
- [ ] Residual risks with business owners
- [ ] Budget and schedule consolidated
- [ ] Clear decision requested: accepted, with risk card, rejected
- [ ] Follow-up actions with owners and due dates

PART 2 · SURVIVING

Surviving the gates: six reflexes

Arrive early

Involve security from the business need: a gate is prepared, not endured.

Record

Every accepted risk becomes an exception, signed, dated and reviewed on time.

Document

A clear, dated security architecture file is worth more than ten meetings.

Follow up

An yellow light commits to an action plan: check it at the next gate.

Prioritize

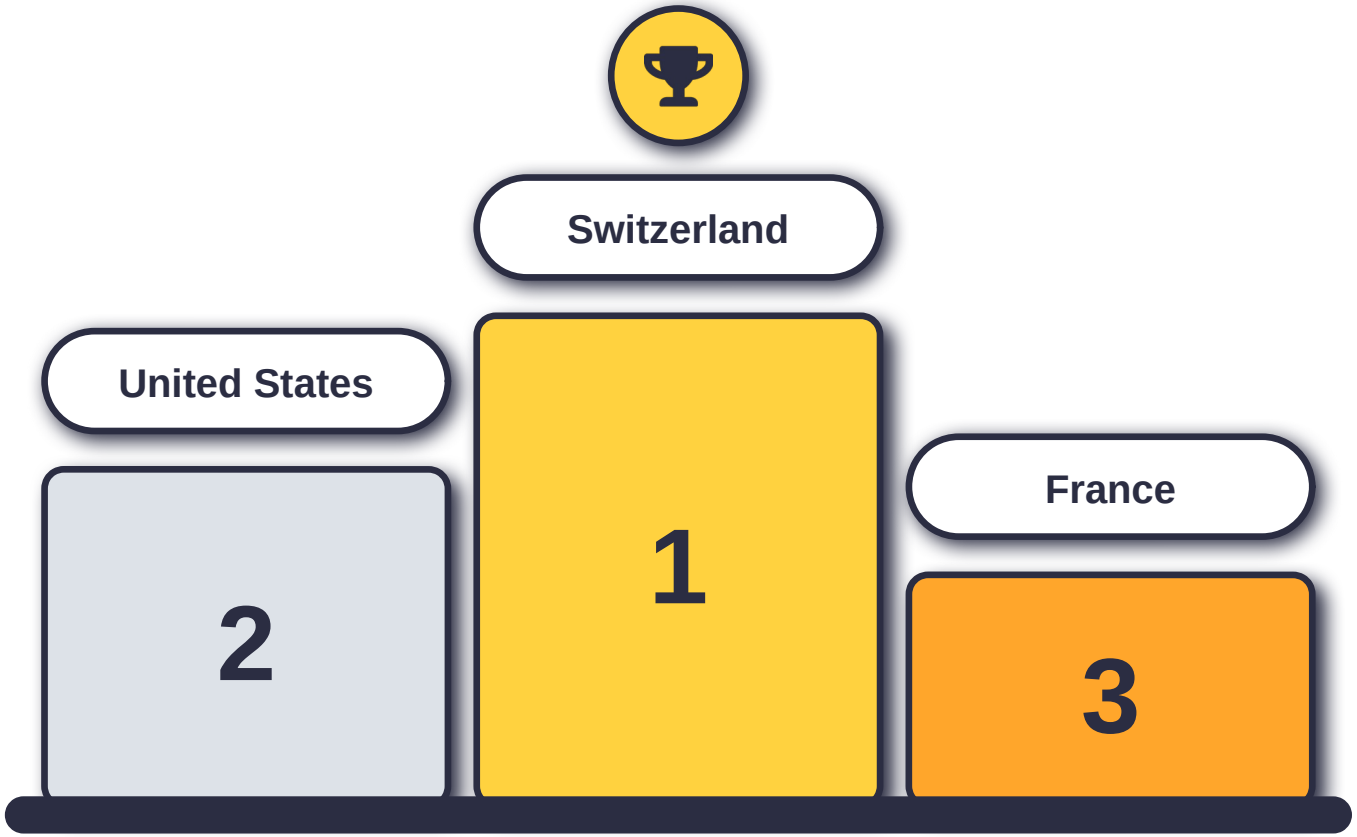
Tell blocking from major and minor: not everything is a red light.

Translate

Talk business impact, not only vulnerabilities: that is how you get heard.

PART 2 · FIELD EXPERIENCE

DGF maturity: my ranking by country



A personal ranking, based on the organizations I have worked with in each country. Not a formal study: digital governance maturity as I experienced it in the field.

4 Singapore

5 Belgium

6 United Arab Emirates

PART 2 · FIELD EXPERIENCE

DGF maturity: my ranking by industry

MOST MATURE

Insurance

1

TIED

Finance / Banking

Pharmaceutical

2

Defense

3

Bottom of the pile…

Retail

4

Same principle as for countries: a personal opinion, based on the large enterprises I have worked with in each industry.

PART 2 · FOR THE REMAINING 70%

Setting up a DGF in 90 days

1 Foundation

Days 1 to 30

- Appoint a DGF owner, with a mandate from management
- Define the four gates and the body that holds each one
- Write the initial assessment form (ten blocks)
- Choose three pilot projects, one per path

2 Tool and process

Days 31 to 60

- Put the tool in place: one ticket per project, one status per gate (ServiceNow, Jira or equivalent)
- Publish the checklist per gate and the opinion template
- Train project managers and Procurement
- Review the pilots, measure the time spent

3 Steering

Days 61 to 90

- Measure: lead times, distribution of lights, exceptions
- Adjust the form and the checklists with feedback
- Roll out to all projects, with no "urgent" exception
- Monthly committee reviewing exceptions and indicators

Without a dedicated application, the DGF lives in emails and spreadsheets. Start with the tool, however simple: one ticket per project, one status per gate, one date per decision.

03

Procurement & Security

Interaction between the Procurement team and the Security & Architecture team

Access the contract from day one

Principle

When purchasing a new platform, the Security & Architecture team must have access to the contract, its annexes and the vendor's answers to carry out a first assessment, before signature.

Why so early?

- A requirement missing from the contract cannot be enforced against the vendor
- Renegotiating after signature is expensive and takes months
- The Security & Architecture gate must confirm the contract, not discover it

Procurement team ⟷ **Security & Architecture**

→ contract, annexes, vendor answers

← first assessment, requirements, points to negotiate

1 Before the RFP

Procurement includes the security requirements in the specifications: standard security annex and vendor questionnaire.

2 During negotiation

Security & Architecture reads the contract, the annexes and the answers, then delivers a first assessment: green, yellow or red light, and the points to negotiate.

3 Before signature

The selected requirements are written into the contract, gaps become formalized exceptions; the file then moves to the Security & Architecture gate.

Points to investigate (1/2): data and solution

Data

- Data location and residency: country, region, hosting provider
- Data ownership and the uses the vendor allows itself
- Segregation between customers (multi-tenant) and supported classification
- Sub-processors and transfers outside the European Union

Solution security

- Authentication: SSO (SAML / OIDC), MFA, role management
- Encryption in transit and at rest, key management (BYOK)
- Logs exportable to the SIEM, administrator traceability
- Penetration test reports and vulnerability fix lead times

Compliance and certifications

- ISO 27001, SOC 2 Type II, industry certifications (HDS, PCI DSS)
- Data processing agreement (GDPR) and standard contractual clauses
- Industry requirements: DORA, NIS2
- Real scope of the certifications: which entity, which service?

Continuity and resilience

- Availability SLA, committed RTO and RPO
- Backups, recovery and continuity plans tested
- Dependencies: underlying hosting provider, regions, third-party services
- Penalties and support commitments

Points to investigate (2/2): vendor and contract

Incident management

- Notification deadline and process (72 h under GDPR)
- Named security contact at the vendor
- Transparency: post-incident reports and remediation plan
- Vendor cyber insurance

Vendor

- Financial health and longevity of the software vendor
- Legal location and applicable laws (sovereignty, CLOUD Act)
- Subcontracting chain and supply chain security
- Roadmap and commitments on the product

Contractual clauses

- Enforceable security annex: requirements and security assurance plan
- Right to audit and access to evidence (reports, certificates)
- Responsibilities, limitation of liability, penalties
- Confidentiality and intellectual property

Reversibility and end of contract

- Data export formats and lead times
- Certified deletion of data at end of contract
- Migration assistance and exit costs
- Source code escrow for critical software

04

IT & Security

Interaction between the IT team and the Security & Architecture team

One architecture, two viewpoints

Principle

The IT gate and the Security & Architecture gate review the same target architecture: the IT department checks consistency, ability to operate and cost; Security checks risks and compliance. A single architecture file must serve both.

Why work together?

- Contradictory requirements between the two gates block the project
- A file reviewed twice, separately, doubles the lead time
- The IT department's standards are the best place to anchor security

1 — Upstream: shared standards

Security & Architecture publishes its requirements and patterns in the IT department's standards catalog: teams design with them, not against them.

2 — During design: a joint review

A single architecture file, reviewed together; gaps are arbitrated during design, not discovered at the gate.

3 — At the gate: two consistent opinions

Both gates rely on the same file; the conditions set by one become actions tracked by the other.

IT gate: promoting the existing framework

Foundation already in place

SOC

Monitoring and incident response: the project plugs into it

AD

Directory and identities: accounts, groups, authentication

SIEM

Log collection: every platform sends its logs there

What the IT gate promotes

- Reuse the existing foundation before building anything
- Raise the constraints to the project at scoping, not at the gate
- Turn them into architecture file requirements, which Security & Architecture integrates from design onwards

What the project provides in return

- Flow diagram: ports, protocols, direction, volumes
- Account, role and permission needs, requested via ServiceNow
- DNS names and certificates required, with their expiry dates
- Logs to send to the SIEM, events for the SOC to monitor
- Network needs: sites, bandwidth, link to the cloud

Constraints to raise

details on the next slide →

DMZ · MPLS · Direct Connect · DNS · PKI · Certificates · Accounts · Tools · …

PART 4 · IT & SECURITY

Constraints the IT department raises to the project

DMZ

All external exposure goes through the DMZ: no direct access from the Internet to the internal network.

MPLS

Inter-site flows use the MPLS network, with its bandwidth, rules and connection lead times.

Direct Connect / ExpressRoute

Flows between the corporate IT and the cloud go through the dedicated private link, not the Internet.

DNS management

Zones and domain names managed by the IT department: no parallel DNS, registration through the existing process.

PKI management

Certificates issued by the corporate PKI: no self-signed certificates, no unvalidated external authority.

Certificate management

Request, renewal and expiry tracking through the IT department's process: no certificate forgotten in production.

Account creation (ServiceNow)

Every account, access or permission request goes through the ServiceNow catalog: approval and traceability.

Mandated tools (Jira, etc.)

Tickets, requests and project tracking in the IT department's tools, Jira first: no parallel tool.

And the others…

Outbound proxy, IP addressing plan, naming conventions, backups, monitoring…

Topics to handle together (1/2): foundation and access

Mapping and standards

- Application and flow mapping, kept up to date
- Catalog of authorized… and forbidden technologies
- Secure architecture patterns, ready to reuse
- Obsolescence management: versions and end of support

Network and hosting

- Segmentation, zones and filtering between environments
- Internet exposure: reverse proxy, WAF, API gateway
- Hosting: cloud, on-premises, regions and location
- Remote access and interconnections with partners

Identities and access

- Single identity repository, SSO and MFA
- Privileged accounts (PAM) and service accounts
- Account life cycle: creation, review, removal
- Segregation of duties and least privilege

Data and integrations

- Classification applied to storage and flows
- Encryption and key management (KMS, HSM)
- APIs, file exchanges, ETL: authentication and logging
- Backups, retention and data purge

Topics to handle together (2/2): operations and governance

Operations and monitoring

- Centralized logging (SIEM) and alerts
- Vulnerability and patch management, with deadlines
- Hardening and baseline configurations
- Change management: security in the loop

Continuity

- Recovery and continuity plan, RTO and RPO
- Backups tested and restorations rehearsed
- Redundancy and critical dependencies identified
- Joint crisis exercises

Life cycle and debt

- Decommissioning: data, access, DNS, certificates
- Technical debt and security debt tracked together
- Exceptions with a due date and an owner
- Periodic review of architectures in production

Joint governance

- Architecture committee with security at the table
- Shared exception register
- Common indicators: standards covered, vulnerabilities, overdue exceptions
- Clear responsibilities: who decides, who executes, who controls

05

Business & Security

Interaction between the business lines and the Security & Architecture team

Business as risk owner

Principle

The business expresses the need, owns the data and carries the risk. Security does not accept a risk on its behalf: it sheds light on it, proposes options with their cost, and gets the owner to sign.

Why?

- Without criticality or RTO / RPO expressed, security invents requirements… and gets them wrong
- A risk accepted by security commits nobody; accepted by the business, it has an owner
- A business that understands the impact arbitrates better than the committee

1 At scoping: a one-hour workshop

Criticality, RTO and RPO, data classification, users and processes, acceptable reversibility: everything that feeds the file comes from the business.

2 During the assessment: be pragmatic

Three protection levels with their cost and lead time; the business chooses knowingly, security recommends.

3 At the gate: a signature

Residual risks are accepted by the business owner, with a risk card, a review date and dated actions.

PART 5 · BUSINESS & SECURITY

What we ask of the business

1 Criticality and tolerance to outages

WHAT WE EXPECT

The criticality of the service for the business, expressed as RTO and RPO, and the sensitive periods: month-end close, sales events, payroll, campaigns.

WHAT FAILS

The business answers "it's critical" to everything, with no RTO. Result: a 99.99% availability requirement for a tool used twice a month, and nothing for payroll.

2 Data: nature, classification, owner

WHAT WE EXPECT

The data processed, its classification, its owner, the people concerned, and what the business agrees to let go to a vendor.

WHAT FAILS

"It's only planning data." It contains salaries per person, extracted from payroll. The "internal" classification was checked by default.

3 Users, roles and processes

WHAT WE EXPECT

Who uses the service, with which roles, in which process; who approves, who administers on the business side; the contractors and partners involved.

WHAT FAILS

The file announces ten finance users. At go-live, two hundred sales reps and three contractors have access: the real process had never been described.

4 Acceptance of residual risks

WHAT WE EXPECT

A named business owner, who understands the impact, signs the risk card and commits to the actions and the review date.

WHAT FAILS

The risk card is signed by the project manager, who leaves three months later. The risk no longer has an owner; the review never happens.

06

Reviewing a file

First pass: read and qualify an architecture file with the CIA+TN grid

PART 6 · REVIEWING A FILE

Compass: CIA+TN

Everything is assessed through five criteria. Each section of the file, each question asked and each opinion given relates to one of them.

C

Confidentiality

Only authorized people access the information.

QUESTION

Who can read what, and how do we prevent everyone else?

MEASURES

Classification, encryption, access control, segregation

I

Integrity

Information is neither altered nor destroyed without authorization.

QUESTION

How do we know the data is accurate and complete?

MEASURES

Signatures, consistency checks, backups, change management

A

Availability

A as in Availability

Information and the service are accessible when needed.

QUESTION

What happens if it goes down, and how long can we wait?

MEASURES

Redundancy, recovery, RTO / RPO, service commitments

T

Traceability

We know who did what, when, and we can reconstruct it.

QUESTION

Can actions and access be traced afterward?

MEASURES

Logging, SIEM, timestamps, retention

N

Non-repudiation

An actor cannot deny having performed an action.

QUESTION

Can we prove an action really came from its author?

MEASURES

Electronic signature, strong authentication, sealed logs

PART 6 · REVIEWING A FILE

Anatomy of an architecture file

1 Context and business need

What the solution does, for whom, with what criticality

C I **A** T N

2 Scope and actors

Users, administrators, vendors, subcontractors

C I A T N

3 Target architecture

Components, environments, zones: a real diagram

C I **A** T N

4 Flows

Ports, protocols, direction, volumes, encryption

C **I** A T N

5 Data and classification

Nature, classification, location, retention

C **I** A T N

6 Identities and access

Authentication, roles, privileged accounts, life cycle

C I A T **N**

7 Hosting and network

Cloud or on-premises, regions, DMZ, private links

C I **A** T N

8 Operations and monitoring

Logging, SIEM and SOC, backups, patches

C I **A** **T** N

9 Risks, measures and exceptions

Risks identified, measures selected, accepted gaps

C **I** **A** **T** **N**

The dots show the CIA+TN criteria each section must primarily address.

Reading grid: three questions per criterion

C	**Confidentiality**	• Which data, which classification, and who accesses it? • Encryption in transit and at rest: who holds the keys? • Are environments and customers segregated?
I	**Integrity**	• How is data validated, signed or checked? • Who can change the configuration and the code, and how is it traced? • Do backups allow a return to a healthy state?
A	**Availability**	• What RTO and RPO does the business expect? • Are redundancy, recovery and critical dependencies described? • What happens if the vendor or the cloud link goes down?
T	**Traceability**	• Which events are logged, where, and for how long? • Do the logs go to the SIEM and the SOC? • Are administrator and service account actions traced?
N	**Non-repudiation**	• Are sensitive actions signed or strongly authenticated? • Reliable timestamps, logs protected against modification? • Can we prove afterward who approved what?

Confidentiality: what we expect

C

1 Data classification and location

WHAT WE EXPECT

The classification of every data item processed (internal, confidential, critical), its hosting and backup location, and the country of the vendor's support.

WHAT FAILS

An AI module added to a SaaS financial planning tool: data classified "critical", hosted abroad, support in a third country. The file says neither where the backups are, nor who at the vendor accesses the data.

2 Encryption and key management

WHAT WE EXPECT

Encryption in transit (TLS 1.2 minimum) and at rest, and above all who holds the keys: the vendor, the customer (BYOK) or an HSM.

WHAT FAILS

"Data is encrypted at rest": yes, with a key managed by the vendor, accessible to its administrators and subcontractors. For critical data, that is not encryption, it is a promise.

3 Access control and segregation

WHAT WE EXPECT

Who accesses what, through which role, from which device; segregation of environments and of customers in a multi-tenant platform.

WHAT FAILS

Ten finance users announced in the file… and a single "administrator" role for everyone, with no distinction between the person entering a budget and the one exporting the whole history.

4 Use of data by third parties and by AI

WHAT WE EXPECT

What the vendor and its subcontractors may do with the data: support, analytics, AI model training; and the contractual evidence.

WHAT FAILS

The box "data is used with AI models" is checked, without saying which. The risk of the vendor's model being trained on the financial data was only handled after the meeting, through a written commitment obtained afterward.

PART 6 · CIA+TN

Integrity: what we expect

I

1 Data validation and control

WHAT WE EXPECT

How incoming and outgoing data is validated: consistency checks, reconciliation with the source, rejections recorded.

WHAT FAILS

An automated budget consolidation from several sources, without reconciliation with the source: a mapping error doubled a line, and nobody saw it before the committee.

2 Change and code management

WHAT WE EXPECT

Who can change the configuration, the rules and the code; review, tests, change traceability, separation between development and production.

WHAT FAILS

Development: "NA, because we use a SaaS". Except that calculation rules are configured by business users directly in production, without review or test environment.

3 Integrity of backups and restoration

WHAT WE EXPECT

Backups protected against modification (immutable or offline), restoration tested, recovery point known.

WHAT FAILS

Backups "handled by the vendor": never tested on the customer side, no evidence of restoration, no commitment on the recovery point. In case of corruption, nobody knows what we would roll back to.

4 Integrity of flows and integrations

WHAT WE EXPECT

Authenticated and signed flows, APIs with integrity checks, protected message queues, rejections and duplicates handled.

WHAT FAILS

A daily export to the ERP through a file dropped on a network share, without signature or check: any user of the share can change the amounts before import.

PART 6 · CIA+TN

Availability: what we expect

A

1 Business requirements: RTO, RPO, criticality

WHAT WE EXPECT

The business has said how long it can wait (RTO) and how much data it can lose (RPO); the declared criticality is consistent with those figures.

WHAT FAILS

Data classified "critical" in the file, but no RTO or RPO. The vendor offers its standard 99.5% commitment and nobody knows whether that is enough for a month-end close.

2 Redundancy and recovery

WHAT WE EXPECT

Redundant architecture, recovery and continuity plans described and tested, backup regions or sites identified.

WHAT FAILS

A SaaS service hosted in a single region, without a contractual backup site. The file states "high availability ensured by the vendor", with no supporting document.

3 Dependencies and links

WHAT WE EXPECT

The list of critical dependencies: hosting provider, private link, identity provider, third-party APIs; and what happens if one of them goes down.

WHAT FAILS

The application only works through the private link to the cloud. The day it goes down for maintenance, the emergency access over the Internet was never planned, nor allowed in the DMZ.

4 Commitments and penalties

WHAT WE EXPECT

Availability SLA, support, recovery lead times and penalties written into the contract, with a shared measurement method.

WHAT FAILS

The 99.9% SLA only covers "the platform": neither the ERP integration nor the AI module added as an option. The service the business actually uses has no commitment at all.

PART 6 · CIA+TN

Traceability: what we expect

T

1 Logging of access and actions

WHAT WE EXPECT

Which events are logged: logins, sensitive actions, exports, configuration changes; by whom, when, from where.

WHAT FAILS

The file states "logging: vendor console". When asked, the vendor specifies: viewable for thirty days, no export. After that, nothing.

2 Export to the SIEM and the SOC

WHAT WE EXPECT

Logs go to the company's SIEM (API, connector, SFTP), in a usable format and within a known delay; the SOC knows what to monitor.

WHAT FAILS

No export planned in the contract: the SOC will never see an exfiltration of the budget history. The blocking condition was only raised at the gate, too late for the negotiation.

3 Log retention and protection

WHAT WE EXPECT

Retention aligned with obligations (often one year), logs protected against modification and deletion, including by administrators.

WHAT FAILS

Logs are stored in the same database as the data, editable by the platform administrator. A malicious administrator erases their tracks with the very rights they just used.

4 Technical accounts and administrators

WHAT WE EXPECT

The actions of service accounts, managed identities, AI agents and vendor administrators are traced and attributable.

WHAT FAILS

Vendor support accesses the data with a generic account shared by its whole team. Impossible to know which person viewed what.

Non-repudiation: what we expect

N

1 Strong authentication for sensitive actions

WHAT WE EXPECT

High-stakes actions (budget approval, transfer, permission change) require strong authentication or re-authentication.

WHAT FAILS

Budget approval is one click away, on a session left open since morning on a shared desk in an open-plan office. "I approved? That wasn't me."

2 Signature and timestamping

WHAT WE EXPECT

Electronic signature of documents and transactions with evidential value, reliable timestamps from a trusted time source.

WHAT FAILS

Financial exports are spreadsheet files without signature or hash. Two versions circulate with different figures, and nothing indicates which one was presented to the committee.

3 Individual attribution

WHAT WE EXPECT

Every action is attributable to a person: named accounts, no generic accounts, service accounts dedicated to a single use.

WHAT FAILS

Stores log in with one account per store, known to the whole team. A fraudulent discount went through the register: impossible to say who entered it.

4 Evidence kept and enforceable

WHAT WE EXPECT

Evidence (signed logs, acknowledgments, approvals) is kept intact for the legal period and can be produced in case of dispute or audit.

WHAT FAILS

In a dispute with a vendor over an order, the company can only produce screenshots: the approval logs were purged after ninety days.

Warning signs

Flows "to be defined later"

Criteria at risk: **C** I A **T** N

A shared service account, a password in the code

Criteria at risk: **C** I A T **N**

Sensitive data without classification

Criteria at risk: **C** I A T N

No flow diagram, just boxes and arrows

Criteria at risk: **C** **I** **A** **T** **N**

"We'll look at backups in production"

Criteria at risk: C **I** **A** T N

A self-signed certificate, the PKI bypassed

Criteria at risk: **C** I A T **N**

Local logs, never exported

Criteria at risk: C I A **T** **N**

An exception without a due date or an owner

Criteria at risk: **C** **I** **A** **T** **N**

07

Technical assessment

DGF questionnaire and technical deep dives

PART 7 · TECHNICAL ASSESSMENT

Initial assessment form

1. Context and business requester
2. Related documents: HLD, ServiceNow request, Procurement approval
3. **Compliance (GRC)**
4. **Data protection: exposure factors**
5. **Digital service overview**
6. Architecture
7. SDLC and operations
8. Cyber risks and red flags
9. Next steps and follow-up actions
10. Risk status

Blocks 3, 4 and 5 are detailed on the right; blocks 6 to 10 each have their own slide.

3 Compliance (GRC)

- Already under contract with this vendor?
- SOC 2 report available?
- ISO 27001 certified at the date of the file?
- ISO 27017 certified (cloud services)?

4 Exposure factors

- Exposes services to the Internet?
- Users outside the protected environment?
- Critical data: confidentiality, integrity, payments?
- Subject to local regulation (country laws)?
- Changes the existing technical protection?
- Data used with AI models?

5 Digital service overview

- Purpose, business domain, type of information
- Classification: public, internal, confidential, restricted, critical
- Front end: web, thick client, terminal, REST
- Managed or unmanaged devices; internal, external, B2B, federated users
- Number of users and location of the service

Architecture block of the form: twelve questions

Twelve questions and expected answers

1. **Overview** — a diagram, not a sentence
2. **On-premises components** — servers, connectors, synchronization agents
3. **Cloud components** — services, regions, tenants
4. **Hosting type** — PaaS, SaaS or IaaS: different responsibilities
5. **Databases** — engine, location, encryption, backups
6. **Interfaces** — every flow: source, target, protocol, port
7. **API gateway** — which one, authentication, quotas
8. **Proxy changes** — rules to add, allowed domains
9. **Email gateway** — the company's, or another one: why?
10. **Authentication** — SSO, MFA, federation… or local accounts?
11. **Authorization and roles** — the matrix of roles and permissions
12. **Regions of use** — countries, applicable laws, latency

SEEN IN A FILE

An AI module added to a SaaS financial planning tool, data classified "critical". The block is filled in: hosting "SaaS", authentication "platform accounts", interfaces "API", regions "worldwide". Four short answers, four questions: why not SSO? which APIs, with which authentication? which countries, which laws? The file goes back with one action: provide the HLD and its security concepts.

To attach to the file

- HLD with the security concepts
- Flow diagram and matrix
- Matrix of roles and permissions
- List of subcontractors and locations

PART 7 · TECHNICAL ASSESSMENT

SDLC and operations

1 Software development life cycle (SDLC)

WHAT WE EXPECT

Who develops (in-house, contractor, vendor), where the code is, how it is reviewed; for a SaaS, the evidence of the vendor's secure life cycle: SOC 2, penetration test report.

WHAT FAILS

Development: "NA, because we use a SaaS". But the vendor does develop: rather than accepting the answer, ask for its latest penetration test report and its SOC 2. Nobody did.

2 Deployment and environments

WHAT WE EXPECT

Automated deployment pipeline, separate environments (development, test, production), secrets in a vault, not in the configuration.

WHAT FAILS

Manual deployment from a developer's workstation, secrets in the configuration file, no test environment: every update goes straight to production.

3 Security testing in the life cycle

WHAT WE EXPECT

Static code analysis (SAST), dynamic testing (DAST), dependency analysis, penetration test before go-live for an exposed or critical service.

WHAT FAILS

No security testing in the life cycle, and the box "penetration test required: no" is checked for a service exposing critical data. The box was checked by the requester.

4 Secure operations

WHAT WE EXPECT

Patches with deadlines by criticality, maintenance windows, monitoring and logging connected, documented operating procedure.

WHAT FAILS

"Updated under the contract": with no fix deadline for critical vulnerabilities, no announced window, no evidence. The contract says "regularly".

PART 7 · TECHNICAL ASSESSMENT

Secrets management

1 No secrets in the code

WHAT WE EXPECT

No password, API key, token or private certificate in the code, the configuration or the repository history; automatic secret scanning on every commit.

WHAT FAILS

A SQL connection string with its password in the configuration file, pushed to the repository two years ago. Removed since… but still in the Git history, cloned by every successive contractor.

2 A vault for secrets

WHAT WE EXPECT

Secrets live in a vault (Azure Key Vault, AWS Secrets Manager, HashiCorp Vault); the application reaches it through a managed identity or a role, never with a secret to fetch a secret; access is logged.

WHAT FAILS

The vault exists, but the application connects to it with a client secret stored… in the configuration file. The vault no longer protects anything: whoever reads the configuration reads the whole vault.

3 Rotation and life cycle

WHAT WE EXPECT

Scheduled rotation of keys, service passwords and certificates; immediate revocation when someone leaves or after an incident; one owner per secret.

WHAT FAILS

Seen in an audit: privileged service accounts with a "never expires" password, unchanged since 2011, known to three contractors who have since left. Nobody knows where it is still used.

4 Pipelines, images and tickets

WHAT WE EXPECT

Encrypted and masked pipeline variables; no secret in build logs, container images, .env files, tickets, wikis or chat.

WHAT FAILS

The database administrator password travels through a ticket, then a chat thread "to go faster". It is also in the Docker image, in clear text, in an environment variable.

PART 7 · DEEP DIVE

Non-production environments

1 Test data

WHAT WE EXPECT

No production data in test without anonymization or masking; synthetic data sets; if real data is indispensable, the same level of protection as in production.

WHAT FAILS

A full copy of the customer database in the test environment, accessible to the whole development team and a contractor, without logs. That is where the leak came from.

2 Access and isolation

WHAT WE EXPECT

Environments isolated from the production network, named access, no flow from test to production, separate service accounts.

WHAT FAILS

The test server has direct access to the production database "to check a mapping". A test script overwrote real data.

3 Secrets and configurations

WHAT WE EXPECT

Separate secrets per environment, never the same keys in test and production, configurations managed as code and reviewed.

WHAT FAILS

The same vendor API key in development, test and production. Leaked from an intern's laptop, it gave access to production.

4 Temporary environments

WHAT WE EXPECT

Defined lifetime, tagging, automatic shutdown, regular review of "temporary" environments, cost tracked.

WHAT FAILS

A demo environment created for a trade show in 2022 is still running, exposed on the Internet, with the vulnerable version of the time.

PART 7 · DEEP DIVE

Infrastructure as code

1 Code and review

WHAT WE EXPECT

Infrastructure described as code (Terraform, Bicep, CloudFormation), versioned in Git: protected branches, mandatory review, a version tag for each deployment, security in the review of sensitive components.

WHAT FAILS

The code exists, but urgent changes are made in the portal "for now". Six months later, the code no longer matches anything.

2 Infrastructure code analysis

WHAT WE EXPECT

Automatic code analysis in the pipeline (tfsec, Checkov, KICS…): public access, missing encryption or overly broad rules block the deployment.

WHAT FAILS

A network rule open to the whole Internet on the database port, pushed without analysis; the scan would have taken ten seconds to reject it.

3 State files (tfstate) and secrets

WHAT WE EXPECT

Remote state (tfstate) in an Azure storage account or a private S3 bucket: public access disabled, encryption, versioning, locking, access limited to the pipeline identity; never in the repository.

WHAT FAILS

The Terraform state file, with database passwords in clear text, is pushed to the Git repository, readable by every developer. Elsewhere, a public state bucket, without versioning: one deletion and the infrastructure is orphaned.

4 Deployment, drift and compliance

WHAT WE EXPECT

Deployment by the pipeline with a managed identity or OIDC federation, no long-lived key; drift detection between the code and reality; policies (Azure Policy, SCP) as a safety net.

WHAT FAILS

The code says "public access disabled"; someone re-enabled it in the portal. Nobody detects the drift: the code gives false assurance.

PART 7 · DEEP DIVE

Software supply chain

1 Open source dependencies

WHAT WE EXPECT

Dependency analysis (SCA) at every build, pinned versions, known vulnerabilities fixed according to their criticality, watch on abandoned components.

WHAT FAILS

A logging library vulnerable for eighteen months, present in four applications, discovered during an external penetration test. Nobody had the list of dependencies.

2 Software bill of materials (SBOM)

WHAT WE EXPECT

A bill of materials (SBOM) generated automatically for each version, and required from the vendor for purchased software.

WHAT FAILS

During a worldwide alert on a component, the company takes three weeks to find out whether it is affected: no internal SBOM, no answer from the vendor.

3 Artifact signing and integrity

WHAT WE EXPECT

Artifacts, images and packages signed and verified at deployment; private registry; traceable build provenance.

WHAT FAILS

The deployment downloads a binary from a public repository without verifying its signature. The day the repository is compromised, production installs the tampered version.

4 Contractor and vendor access

WHAT WE EXPECT

Named, temporary accounts for contractors, access through PAM, review at the end of each engagement; for the vendor, traced support access that can be enabled on demand.

WHAT FAILS

Seen in an audit: an external partner with full, permanent administrator rights, without just-in-time access. Its compromise would be the compromise of the whole tenant.

PART 7 · DEEP DIVE

Containers and Kubernetes

1 Images and registry

WHAT WE EXPECT

Maintained, minimal base images, vulnerability scan at every image build, private registry with signing, no "latest" image in production.

WHAT FAILS

An image built on a three-year-old base, pulled from a public registry, never scanned: forty critical vulnerabilities at the first scan… run after go-live.

2 Execution and privileges

WHAT WE EXPECT

Non-root containers, read-only file system, reduced capabilities, resource limits; no privileged container, no Docker socket mount.

WHAT FAILS

A "privileged" container to "make a monitoring agent work": a container escape gives the node, then the whole cluster.

3 Admission policies and cluster RBAC

WHAT WE EXPECT

Admission policies (no root, signed images, mandatory labels), least-privilege cluster RBAC, namespaces per application and per environment.

WHAT FAILS

Every developer is cluster-admin "until the roles are defined". A year later, a compromised developer account wipes a production namespace.

4 Cluster network and secrets

WHAT WE EXPECT

Network policies between namespaces (deny by default), ingress through a gateway with TLS, secrets externalized in a vault, never in clear text in the manifests.

WHAT FAILS

No network policy: every pod talks to every other. Secrets are base64-encoded in the Git repository manifests, "encrypted" according to the file.

PART 7 · DEEP DIVE

Vulnerabilities and hardening

1 Baseline configurations

WHAT WE EXPECT

Documented hardening (CIS benchmarks, vendor guides) applied automatically, with drift control; default accounts and services disabled.

WHAT FAILS

Servers installed "by default": original local accounts, unnecessary services open, and the same administrator password across the whole fleet because the base image contained it.

2 Scanning and inventory

WHAT WE EXPECT

Up-to-date inventory of all assets (cloud and containers included), authenticated and regular vulnerability scanning, results reconciled with the inventory.

WHAT FAILS

The scan covers the known servers; the virtual machines the project created in a separate subscription are not in it. They went unpatched for a year.

3 Fix deadlines

WHAT WE EXPECT

Deadlines by criticality written into the procedure (for example critical within 7 days, high within 30), exceptions recorded, monthly tracking of delays.

WHAT FAILS

The SaaS contract promises "regular updates". A critical vulnerability, actively exploited, took four months to be fixed, without the company knowing.

4 Server protection and detection

WHAT WE EXPECT

EDR on every server and administration workstation, connected to the SOC; file integrity monitoring on critical systems.

WHAT FAILS

EDR is deployed on workstations, not on servers "so as not to degrade performance". The attacker spent six weeks on a file server without being seen.

PART 7 · DEEP DIVE

Penetration tests

1 Scope and method

WHAT WE EXPECT

Written scope (applications, APIs, infrastructure, mobile, identities), method (black, gray, white box), business scenarios included, signed rules of engagement.

WHAT FAILS

The test covers the login page only, black box, one day. The report is green; the API exposing the data was not in scope.

2 Frequency and triggers

WHAT WE EXPECT

Test before any exposed or critical go-live, then yearly, and at every major change; the vendor's test required, and read.

WHAT FAILS

The last test is three years and two major versions old. The file has "penetration test: done" checked, with no date.

3 Remediation and retest

WHAT WE EXPECT

Every finding has an owner and a deadline by criticality; retest of critical and high findings before go-live.

WHAT FAILS

The report lists four critical vulnerabilities. Six months later, nobody has dealt with them: the report was filed "for information".

4 Reports and confidentiality

WHAT WE EXPECT

Reports stored in a restricted space, controlled distribution, aggregated results reported to governance, trends tracked from one test to the next.

WHAT FAILS

The full report, with detailed exploits, travels by email all the way to the development contractor and its subcontractors.

PART 7 · IAM

IAM: the point that 90% assess badly

FIELD FINDING

90 %

of the people who review a file do not properly assess the identity and access part (IAM).

Why?

Because "who has access to what" hides in seven different places. We check SSO and MFA… and stop there.

Personal finding, based on the files I have reviewed or re-read.

Seven points to assess

Azure RBAC roles and Entra ID roles	Who administers what, and at which scope
Managed identities	Resources talking to services without a secret
Delegated permissions	An application acting on behalf of a user
Application permissions (Graph API)	An application acting on its own, across the whole tenant
AI agent identity	The agent has its own identity… or borrows someone else's
Conditional Access	Conditions imposed at sign-in, service by service
Break-glass account	The emergency account that exists, is excluded, monitored and tested

A proper IAM assessment answers, for each point: who, what, which scope, which consent, under which conditions… and how we regain control.

PART 7 · IAM

Azure permissions (1/2): the foundation and identities

Azure RBAC roles and Entra ID roles

Two separate planes: Azure roles (Owner, Contributor, Reader…) grant rights on resources, at a given scope (subscription, resource group, resource); Entra ID roles (Global Administrator…) manage the directory.

WHEN TO USE IT

To grant rights to people: at the narrowest possible scope, through groups, with just-in-time activation (PIM) for privileged roles.

CLASSIC MISTAKE

The file states "the project team is Contributor on the subscription". Twenty people can change every production resource, including other projects', and one of the accounts is also Global Administrator. Nobody saw it.

Managed identities

An Entra ID identity attached to an Azure resource (virtual machine, function, web app). Azure manages the secrets: nothing to store, nothing to rotate.

WHEN TO USE IT

Whenever an Azure resource must call an Azure service: Key Vault, Storage, SQL, Graph. Prefer a user-assigned identity, shared with discernment.

CLASSIC MISTAKE

An Azure Function gets a managed identity "to read a secret"… with the Contributor role on the whole resource group. The reviewer approves: "no stored secret, so it's safe". The scope, though, was never looked at.

Delegated permissions

The application acts on behalf of a signed-in user, within the limits of their rights: its effective permissions are the intersection of the user's rights and the permissions consented to the application.

WHEN TO USE IT

For any application with a user in front of the screen: portal, mobile app, add-in. Admin consent as soon as the permission is sensitive.

CLASSIC MISTAKE

The file announces "the application only reads the user's calendar". But user consent is open to the whole tenant: anyone can grant Mail.Read or Files.ReadWrite to an unknown third-party application. Nobody checked.

Azure permissions (2/2): applications and agents

Application permissions (Graph API)

The application acts on its own, without a user, with a secret or a certificate. A Graph application permission applies to the whole tenant: Mail.Read means every mailbox.

WHEN TO USE IT

For processing without a user: synchronization, automation, overnight jobs. Always with admin consent, restricted by an access policy (Exchange, SharePoint) where possible.

CLASSIC MISTAKE

An HR connector requests User.ReadWrite.All "to update a few attributes". The reviewer reads "read-write on users" and approves. In reality the application can change every account in the tenant, with a secret sitting in a configuration file.

AI agent identity

An agent (Copilot Studio, Azure AI Foundry or in-house) must have its own identity in Entra ID, distinct from the developer and the user, with its own permissions and its own life cycle.

WHEN TO USE IT

Whenever an agent accesses data or triggers actions: tools, connectors, APIs. Minimal scope, permissions reviewed at every new tool, actions logged.

CLASSIC MISTAKE

The "support assistant" agent runs with the developer's service principal, which has broad Graph application permissions. A user asks it "show me the executives' tickets"... and it finds them. The file merely said: "the agent uses the Graph API".

Conditional Access

Policies applied at sign-in, service by service: this user, this device, this location, this risk, to this application → require MFA, a compliant device... or block.

WHEN TO USE IT

To impose conditions on sensitive services: Graph, Azure portal, business applications. Including technical identities (Conditional Access for workload identities).

CLASSIC MISTAKE

The file states "MFA mandatory via Conditional Access". The policy excludes service accounts and administrators "to avoid lockouts" and only targets the applications listed when it was created: the new platform is not among them. MFA is mandatory for nobody who matters.

AWS permissions (1/2): identities and roles

IAM policies, users and roles

IAM policies (JSON) attached to users, groups or roles define what is allowed: actions, resources (ARNs), conditions. A role is assumed temporarily, with short-lived credentials.

WHEN TO USE IT

Roles for everything human (through Identity Center) and for applications; no IAM user with a permanent access key, except for a recorded and rotated exception.

CLASSIC MISTAKE

The file plans a "service" IAM user with a permanent access key, copied into three applications. Leaked in a public repository, it was used to spin up crypto-mining machines: the bill doubled before anyone noticed.

Roles attached to resources

AWS services (EC2, Lambda, ECS, EKS) assume a role attached to the resource: temporary credentials provided by the service, nothing to store, automatic rotation.

WHEN TO USE IT

Whenever a resource calls an AWS service: S3, KMS, Secrets Manager, DynamoDB. One role per function, scope limited to the necessary ARNs.

CLASSIC MISTAKE

An instance role with "AdministratorAccess" attached to the whole EC2 fleet "to make it work". An injection on a single web application gives full administration of the account.

Permission boundaries

A permission boundary caps what a role can obtain, whatever its policies: it allows delegating role creation without losing control.

WHEN TO USE IT

To let teams create their roles within a perimeter: resources prefixed with the project name, allowed regions, forbidden actions.

CLASSIC MISTAKE

Role creation is delegated without a permission boundary: a team creates an administrator role for itself. Or, conversely, the boundary cannot be written for lack of a naming convention.

PART 7 · IAM

AWS permissions (2/2): organization and data

Organizations and SCPs

Service control policies (SCPs) apply to entire organizational units: they deny, they never grant. Nothing bypasses them, not even the root account of a member account.

WHEN TO USE IT

As guardrails of the landing zone: forbidden regions, forbidden services, protection of CloudTrail and GuardDuty, no leaving the organization.

CLASSIC MISTAKE

A production account created outside the organization "for a pilot": no SCP, no centralized CloudTrail. It has hosted the e-commerce site for two years.

Human access: Identity Center and root

IAM Identity Center federates the directory (Entra ID) to all accounts with permission sets; root accounts are emergency accounts, without access keys, protected by hardware MFA.

WHEN TO USE IT

For any human access to the console or the command line: never a local IAM user, never root for daily work, MFA everywhere.

CLASSIC MISTAKE

The root account is used for day-to-day administration, with a shared password and no MFA. It is the emergency account... and the everyday account.

Resource policies and KMS

Bucket, queue and KMS key policies add to IAM policies: effective access is the intersection. "Block Public Access" closes the door at account level.

WHEN TO USE IT

Every sensitive bucket has its policy (TLS mandatory, allowed principals) and every KMS key its own policy; public access is blocked at account level.

CLASSIC MISTAKE

A bucket with a policy open to any principal "so the vendor can drop its files". The exports can be listed from the Internet; Block Public Access was not enabled.

Classic omission: the break-glass account

Break-glass account

An emergency administration account, cloud-only, that lets you regain control when everything else is blocked: misconfigured Conditional Access policy, MFA provider outage, federation down, last administrator gone.

RULES

- At least two accounts, cloud-only: neither federated nor synchronized
- Excluded from every Conditional Access policy
- Long password in a physical safe, or a dedicated FIDO2 key
- Permanent Global Administrator role: no just-in-time activation
- Monitored: every sign-in triggers an alert to the SOC
- Tested at regular intervals, procedure documented and known

CLASSIC MISTAKE

The new Conditional Access policy requires a compliant device for all administrators. It is enabled on a Friday evening. No administration workstation has been declared compliant yet: nobody can sign in to the portal anymore… and nobody created a break-glass account. The tenant stays locked until a ticket is opened with Microsoft.

Variant: the account exists, but it is synchronized from the on-premises AD and it is precisely the federation that is down. It is useless.

Question to ask the file

Does the file describe the break-glass account: existence, exclusions, secret storage, sign-in alert, date of the last test?

PART 7 · IAM

Seen in an audit (1/2): accounts and processes

Critical	**External partner with full administrator rights** — without just-in-time access (PIM): a supply chain risk across the whole tenant.	C I
High	**No break-glass account** — an administrator uses the Global Administrator role for daily tasks.	C A
High	**SecOps lead without the Global Reader role** — impossible to audit what the IT department and partners deploy: a blind spot.	T
Critical	**MFA not enforced for all users** — the first protection against credential theft is still missing.	C N
Critical	**No geographic restriction** — no Conditional Access policy limiting sign-in to the countries of operation.	C
Critical	**Around forty guests on personal mailboxes** — Gmail, Yahoo, Hotmail… with no life cycle and no owner.	C T
Critical	**982 accounts disabled in Entra, still synchronized from the on-premises AD** — re-enablable if the AD is compromised.	C I
High	**"Cloud-only" accounts created outside the process** — they escape the on-premises AD policies and the ServiceNow workflow.	C T

Findings from a real Entra ID audit carried out in 2026, anonymized: names, accounts and organizations have been removed. The dots show the CIA+TN criteria affected.

PART 7 · IAM

Seen in an audit (2/2): passwords and applications

Critical	**Passwords never changed since 2011-2012** — including on administration accounts: permanent exposure.	C
Critical	**Privileged service accounts with non-expiring passwords** — mail, SAP, file servers: lateral movement guaranteed.	C I
Critical	**Point-of-sale (POS) systems with static credentials** — a target for banking malware, PCI DSS non-compliance.	C I
High	**Shared mailboxes and generic accounts** — password unchanged when employees leave: former staff keep access.	C N
High	**Third-party vendor accounts with static passwords since 2018-2021** — backdoors if the relationship or the vendor is compromised.	C
Critical	**Oversized application permissions** — a migration tool with Directory.AccessAsUser.All, a video conferencing tool with Sites.Read.All.	C
High	**Obscurely named applications with broad permissions** — raw identifiers, impossible to link to an owner.	T
Medium	**Privileged delegated consents never revoked** — an application used once keeps its rights forever.	C

Findings from a real Entra ID audit carried out in 2026, anonymized: names, accounts and organizations have been removed. The dots show the CIA+TN criteria affected.

PART 7 · DEEP DIVE

Hybrid directory: AD and Entra ID

1 Synchronization server, a tier-0 asset

WHAT WE EXPECT

The synchronization server (Entra Connect or Cloud Sync) treated like a domain controller: hardened, isolated, administered from dedicated workstations, monitored.

WHAT FAILS

Seen in an audit: the synchronization server was never audited, and it is administered with ordinary accounts. Whoever controls it controls both directories.

2 Disabled accounts and synchronization scope

WHAT WE EXPECT

Disabled accounts are removed from the AD or excluded from synchronization; scope limited to the useful organizational units; service accounts excluded.

WHAT FAILS

Seen in an audit: 982 accounts disabled in Entra but still synchronized from the AD. If the AD is compromised, the attacker re-enables them… and they come back in the cloud.

3 Passwords and policies, AD and cloud

WHAT WE EXPECT

Documented authentication method, cloud password protection extended to the AD, consistent policies on both sides, cloud MFA for everyone.

WHAT FAILS

Seen in an audit: cloud-only accounts, created outside the process, that escape the AD policies (complexity, lockout, hours). Nobody on the AD side sees them.

4 Privileged accounts, AD and cloud

WHAT WE EXPECT

No synchronized account holds a privileged Entra role; cloud administrators on dedicated cloud-only accounts; AD administrators with no cloud role.

WHAT FAILS

A domain administrator is also Global Administrator, with the same synchronized account: a compromise of the on-premises AD is a compromise of the whole tenant.

PART 7 · DEEP DIVE

External identities and guests

1 Guest life cycle

WHAT WE EXPECT

Invitation by an internal sponsor, justification, end date, automatic deactivation after inactivity, deletion at term; personal mailboxes forbidden.

WHAT FAILS

Seen in an audit: around forty guests on Gmail, Yahoo, Hotmail… with no owner and no end date. Some have been there for years; nobody knows who they are.

2 Access reviews

WHAT WE EXPECT

Quarterly review of guests and their access by the sponsor, removal by default if nobody answers; review of groups and applications open to externals.

WHAT FAILS

No review since the tenant was created. A former contractor, invited in 2019, still has access to the RFP SharePoint site.

3 Entitlement management

WHAT WE EXPECT

Access packages per partner use case: what is granted, to whom, for how long, with approval and automatic expiry.

WHAT FAILS

Every partner request is handled manually by the help desk: different rights for the same need, never removed, and no overall view.

4 Conditional Access and B2B collaboration

WHAT WE EXPECT

Conditional Access policies dedicated to guests (MFA, restrictions), external collaboration limited to allowed domains, invitations reserved to defined roles.

WHAT FAILS

Any user can invite anyone, with no MFA enforced on guests. A compromised guest account was used to phish employees from inside the tenant.

PART 7 · DEEP DIVE

Administrator access

1 Dedicated administration workstations

WHAT WE EXPECT

Administrators work from hardened privileged access workstations (PAW), without email or web browsing, separate from their office workstation.

WHAT FAILS

The tenant administrator does their administration from their office workstation, between two emails. One successful phishing attack on that workstation is worth Global Administrator access.

2 Bastion and protocols

WHAT WE EXPECT

No RDP or SSH exposed; access through a bastion (Azure Bastion, AWS Session Manager) or a PAM solution, with MFA, session recording and logging.

WHAT FAILS

An RDP port open on the Internet "for the duration of the project", protected by a password. Found by an automated scan within hours, it has been attacked continuously ever since.

3 Administration accounts and roles

WHAT WE EXPECT

Administration account separate from the user account, just-in-time activation (PIM) with justification and duration, quarterly review of privileged roles.

WHAT FAILS

Seen in an audit: the Global Administrator role used daily by one person, permanently, with the same account they use for email.

4 Service accounts and automation

WHAT WE EXPECT

Automations use managed identities or dedicated service accounts, least privilege, no interactive sign-in, with an owner and a review.

WHAT FAILS

A Domain Admin service account is used for backups, monitoring and a contractor's scripts all at once. Its password sits in three different tools.

PART 7 · DEEP DIVE

Workstations and devices

1 Managed devices and compliance

WHAT WE EXPECT

Access to data from managed devices (Intune or equivalent), encrypted, up to date, with EDR; compliance is enforced by Conditional Access.

WHAT FAILS

The file has "managed devices only" checked. But the Conditional Access policy does not require compliance: any browser, on any personal PC, reaches the application.

2 Personal devices (BYOD)

WHAT WE EXPECT

If personal devices are allowed: application container (MAM), no local copy of data, remote wipe of the professional part.

WHAT FAILS

Sales reps consult the CRM from their personal phone with the native app: customer data synchronized locally, no passcode, no wipe possible when they leave.

3 Remote access

WHAT WE EXPECT

Remote access through the standard solution (VPN with MFA or Zero Trust access), from a managed device; no individual VPN, no exposed remote desktop.

WHAT FAILS

A contractor reaches the internal network through a dedicated VPN, without MFA, with an account shared by its whole team, opened in 2019 and never reviewed.

4 Device life cycle

WHAT WE EXPECT

Enrollment on arrival, wipe and access removal on departure, device inventory, disk encryption and recovery keys in escrow.

WHAT FAILS

The laptop of an employee who left six months ago is still enrolled and still connected; it holds a local copy of the financial exports.

PART 7 · DEEP DIVE

Point of sale, IoT and industrial systems

1 Credentials and accounts

WHAT WE EXPECT

No default credential, no shared account: one account per device, password rotation, no permanent administrator account on the registers.

WHAT FAILS

Seen in an audit: registers with static credentials, never changed, known to every store employee. An ideal target for banking malware.

2 Segmentation

WHAT WE EXPECT

Registers, IoT and industrial systems in dedicated, filtered networks, with no direct access to the corporate IT or the Internet; PCI DSS for everything that touches payment.

WHAT FAILS

The registers are on the same network as the store's office workstations. A workstation infected through an email reaches the payment terminals within minutes.

3 Updates and hardening

WHAT WE EXPECT

Device inventory, scheduled updates (firmware included), hardening, replacement or strict isolation of obsolete systems that cannot be patched.

WHAT FAILS

A third of the register fleet runs on a system out of support for five years, "because the register software doesn't work on the new version".

4 Monitoring and maintenance access

WHAT WE EXPECT

Register and PLC logs connected to the SOC; vendor maintenance access through a bastion, traced and time-limited.

WHAT FAILS

The register vendor reaches every store through a remote control tool, with a single account, without logs or limits.

PART 7 · DEEP DIVE

Cloud resource organization

1 One resource group per project (Azure)

WHAT WE EXPECT

All the project's resources in a single resource group per environment. It is the boundary for RBAC roles, policies, locks and life cycle… and for billing: one cost per project, readable without any manual tagging.

WHAT FAILS

Resources scattered across three groups shared with other teams: bill impossible to isolate, a Contributor role "for the project" that also covers other people's resources, and nobody knows what to delete at decommissioning.

2 A naming convention

WHAT WE EXPECT

A name built according to a convention: environment, project, type, region, number (prd-crm-vm-weu-01). On AWS, the name serves permissions: a permission boundary restricts a role to resources whose ARN starts with "proj-crm-*".

WHAT FAILS

Without a convention, the permission boundary cannot be written: developers get either access to every resource in the account, or none. The file says "naming: to be defined"; in production, the names are "test", "test2" and "final".

3 Mandatory tags

WHAT WE EXPECT

Every resource carries the mandatory tags: project, owner, environment, classification, cost center. A policy denies the creation of a resource without them.

WHAT FAILS

Half the resources have no "owner" tag. During an incident, nobody knows who to call; during the cost review, a third of the cloud bill is "unallocated".

4 Subscriptions, accounts and landing zone

WHAT WE EXPECT

The project lands in the landing zone: a dedicated subscription or account according to criticality, under the management group (or organizational unit) that inherits the policies and the logging. Nothing in the sandbox, nothing under the root.

WHAT FAILS

Seen in an audit: two legacy subscriptions attached directly to the tenant root, outside the management groups. They escape every landing zone policy… and host production.

PART 7 · DEEP DIVE

Tenant and cloud account hygiene

1 Secure score and recommendations

WHAT WE EXPECT

Defender for Cloud or Security Hub enabled on every subscription and account, score tracked, critical recommendations handled before hosting a project.

WHAT FAILS

The project lands in a subscription whose secure score has been stuck at 30% for two years: no encryption, open ports, logs disabled. The file, meanwhile, is perfect.

2 Policies and guardrails

WHAT WE EXPECT

Azure Policy or SCPs inherited from the landing zone: allowed regions, mandatory tags, enforced encryption, public access forbidden, resource creation controlled.

WHAT FAILS

Seen in an audit: subscriptions attached directly to the tenant root, outside the management groups. No policy applies to them, and they host production.

3 Activity logs and alerts

WHAT WE EXPECT

Activity and audit logs (Activity Log, CloudTrail) enabled everywhere, sent to the SIEM, kept at least one year; alerts on sensitive actions.

WHAT FAILS

CloudTrail enabled in a single region, with seven days of retention. After the incident, impossible to know who created the access key the attacker used.

4 Costs and drifts

WHAT WE EXPECT

Budgets and cost alerts per project, monthly review, detection of orphan resources; a cost drift is also a security signal (crypto-mining, exfiltration).

WHAT FAILS

The cloud bill doubles in a month. Nobody reacts before the month-end close: mining machines had been running for three weeks with an access key leaked in a public repository.

PART 7 · DEEP DIVE

License prerequisites

1 Identities and access

WHAT WE EXPECT

The expected features have their license: PIM, access reviews, entitlement management, identity protection and advanced Conditional Access require Entra ID P2 or equivalent.

WHAT FAILS

The file requires "just-in-time activation via PIM". The tenant only has P1 licenses: the recommendation stays on paper for a year, until the budget is found.

2 Protection and detection

WHAT WE EXPECT

Defender plans (servers, storage, databases, containers) or equivalents enabled on the project's subscriptions; EDR licensed for every server.

WHAT FAILS

Defender for Cloud is enabled "in the free tier": the score shows, but detection on the project's virtual machines does not exist.

3 Logging and retention

WHAT WE EXPECT

Extended audit logs, long retention and SIEM ingestion priced into the project budget, not discovered on the invoice.

WHAT FAILS

Log export to the SIEM is planned in the contract... but the ingestion cost (forty gigabytes per day) has no budget. The export is disabled in the second month.

4 Vendor and tiers

WHAT WE EXPECT

The SaaS tier selected includes the security features requested: SSO, log export, customer-managed encryption, security support, often reserved for higher tiers.

WHAT FAILS

SSO is in the "Enterprise" tier, not in the one purchased. The project starts with local accounts and passwords, "pending budget".

PART 7 · DEEP DIVE

Network and exposure

1 Exposure and DMZ

WHAT WE EXPECT

Every exposed service goes through the DMZ: reverse proxy, WAF, API gateway; no direct exposure of an internal component, rate limiting on APIs.

WHAT FAILS

An API "temporarily" exposed directly from the internal network for a vendor test, never removed, without WAF or rate limiting. Six months later, it is still there.

2 Flow matrix

WHAT WE EXPECT

A matrix of source, destination, port, protocol, direction and justification; "any to any" flows are refused, unused flows closed.

WHAT FAILS

The diagram shows arrows without ports. So as not to block the project, the network team opens "whatever is needed" between the two zones. The matrix was never written.

3 Interconnections

WHAT WE EXPECT

Flows to the cloud use the private link; partners go through a dedicated, filtered and logged zone, never straight into the internal network.

WHAT FAILS

A partner connected by site-to-site VPN straight into the internal network, with access to the whole subnet, "because it's simpler". Its incident becomes ours.

4 Proxy and remote access

WHAT WE EXPECT

Internet egress through the corporate proxy with allowed domains; remote access through the standard solution, MFA and managed workstations.

WHAT FAILS

The server bypasses the proxy with a direct route "to make updates work": an unfiltered, unlogged egress, ideal for exfiltration.

PART 7 · DEEP DIVE

DDoS protection and CDN

1 Protection of exposed services

WHAT WE EXPECT

Public services behind DDoS protection (cloud provider or CDN), known absorption capacity, reaction plan for a volumetric attack.

WHAT FAILS

The e-commerce site is hosted on a direct public IP address, without protection. A ten-minute attack on the first day of a major sale cost a full day of revenue.

2 Managed WAF and anti-bot

WHAT WE EXPECT

WAF with managed, up-to-date rules, bot protection (price scraping, credential stuffing), rate limiting per client.

WHAT FAILS

No bot protection: a competitor scrapes every price each night, and credential stuffing attacks test thousands of customer accounts.

3 Hidden origin

WHAT WE EXPECT

The origin (servers, load balancer) is only reachable from the CDN or the WAF; origin addresses not published; old DNS records cleaned up.

WHAT FAILS

The origin IP address is still directly reachable: the attacker bypasses the WAF by targeting the address, found in an old DNS record.

4 Degraded mode and continuity

WHAT WE EXPECT

Degraded mode planned (waiting page, virtual queue, static content), load tests, protection provider contacts known to the on-call team.

WHAT FAILS

Under attack, the only answer is a server error page showing the application framework version. Nobody knows who to call at the hosting provider.

PART 7 · DEEP DIVE

PaaS services and private access

1 Public access disabled

WHAT WE EXPECT

Storage, databases, vaults and message queues without public access: service firewall enabled, public access disabled, exceptions justified and dated.

WHAT FAILS

A storage account with public access "so the vendor can drop the files". The financial exports there are readable by anyone who guesses the address.

2 Private endpoints

WHAT WE EXPECT

Access to PaaS services through private endpoints (Private Link, VPC endpoints), consistent private DNS, flows that never leave for the Internet.

WHAT FAILS

The file announces "database on a private network". It does have a private endpoint... and public access stayed enabled "just in case". The private endpoint was merely one more access path.

3 Identity rather than keys

WHAT WE EXPECT

Access through identity (Entra ID, IAM roles) rather than account keys or connection strings; keys disabled where possible, otherwise rotated and stored in the vault.

WHAT FAILS

The application reaches storage with the primary account key, copied into four applications. Impossible to rotate it without breaking everything: it has not changed since the account was created.

4 Service protection

WHAT WE EXPECT

Encryption at rest with customer keys if the data requires it, soft delete and purge protection (vaults, backups), locks on critical resources.

WHAT FAILS

A key vault without soft delete or purge protection: a badly written script deletes it, and with it the encryption keys of the production database.

PART 7 · DEEP DIVE

APIs and integrations

1 Gateway and exposure

WHAT WE EXPECT

Every exposed API goes through the API gateway: centralized authentication, TLS, logging, WAF; no API called directly from the Internet.

WHAT FAILS

The vendor's API is called directly from the Internet with a static key in the URL. The key shows up in the proxy logs, the browser and the vendor's logs.

2 Authentication and authorization

WHAT WE EXPECT

OAuth 2 with short-lived tokens and limited scopes, mTLS between systems, authorization checked at every call and not only at login, client secret rotation.

WHAT FAILS

A "lifetime" token generated for the ERP integration, shared with two other integrations "to save time". Revoking it would break three flows; nobody dares.

3 Quotas, throttling and validation

WHAT WE EXPECT

Rate limiting and quotas per consumer, strict input and schema validation, maximum sizes, protection against enumeration.

WHAT FAILS

No rate limiting on the customer search API: a script extracted the whole customer base in one night, at a thousand requests per minute, without triggering an alert.

4 Versions and life cycle

WHAT WE EXPECT

Documented API contract (OpenAPI), managed versions, announced deprecation, inventory of consumers, removal of unused APIs.

WHAT FAILS

A "v1" API abandoned three years ago stays exposed with the old authentication, because nobody knows who still uses it. That is how the attacker got in.

PART 7 · DEEP DIVE

Mobile applications

1 Local storage and data

WHAT WE EXPECT

Sensitive data encrypted on the device, minimal data cached, wiped at logout, nothing in the logs or screenshots.

WHAT FAILS

The application caches the full customer list, in clear text, in a file readable by any other application on an unmanaged phone.

2 Authentication and tokens

WHAT WE EXPECT

SSO through the identity provider, short-lived tokens stored in the secure keychain, controlled refresh; biometrics in addition, never instead.

WHAT FAILS

The access token is valid for a year, stored in the application preferences. A lost phone is worth a year of access.

3 Communication and APIs

WHAT WE EXPECT

TLS mandatory, certificate pinning on critical APIs, the same server-side controls as for the web: the mobile application is not trusted.

WHAT FAILS

Permission checks are done in the mobile application, not on the API. With a proxy, a user changes prices they are not supposed to see.

4 Distribution and management

WHAT WE EXPECT

Distribution through the official stores or the enterprise management tool, minimum version enforced, jailbroken devices refused, application management (MAM).

WHAT FAILS

The internal application is distributed through a direct link to an installation file, without verified signature or minimum version. The 2021 version still works.

PART 7 · DEEP DIVE

Certificates and PKI

1 Issuance by the corporate PKI

WHAT WE EXPECT

Internal certificates issued by the corporate PKI, public certificates by the chosen authority; no self-signed certificate in production, no unknown authority in the trust stores.

WHAT FAILS

A self-signed certificate on the internal reverse proxy "for now". Users have learned to click "proceed anyway"; they do it on a phishing site too.

2 Automated renewal

WHAT WE EXPECT

Automatic renewal (ACME or equivalent) with sufficient lead time, documented fallback procedure, one owner per certificate.

WHAT FAILS

The payment API certificate expires on a Sunday: two hours of online sales outage. It had been installed by hand by a contractor who has since left.

3 Inventory and expiry tracking

WHAT WE EXPECT

Central inventory of all certificates (appliances and cloud services included), alerts at 60, 30 and 7 days, regular review.

WHAT FAILS

No inventory: nobody knows how many certificates exist. They are discovered when they expire, on average one incident a month.

4 Private keys and TLS

WHAT WE EXPECT

Non-exportable private keys, stored in an HSM or a vault, TLS 1.2 minimum with modern cipher suites, HSTS on web services.

WHAT FAILS

The private key of the company's wildcard certificate travels by email between three contractors, in a file protected by the password "1234".

PART 7 · DEEP DIVE

Email and notifications

1 Email gateway

WHAT WE EXPECT

The application's outbound mail goes through the corporate email gateway (or an authorized relay), with authentication, filtering and logging.

WHAT FAILS

The application sends its notifications through a third-party SMTP service set up "in five minutes", with a shared account and no logs. One day, it sends phishing in the company's name.

2 Sending domain authentication

WHAT WE EXPECT

SPF, DKIM and DMARC (in reject mode) on every domain and subdomain used to send, including those delegated to the vendor.

WHAT FAILS

The vendor sends on behalf of the company's domain without being in the SPF or signing with DKIM: legitimate emails end up in the spam folder, and fake ones get through as easily as real ones.

3 Notification content

WHAT WE EXPECT

No confidential data in the body or attachments of automatic notifications; a link to the authenticated application rather than the content.

WHAT FAILS

Budget alerts send the detail of variances per entity, in clear text, to a distribution list that still contains former employees and a contractor.

4 Shared mailboxes and accounts

WHAT WE EXPECT

Shared mailboxes with owners and reviewed access, no shared password, MFA on every mail account, legacy protocols disabled.

WHAT FAILS

Seen in an audit: shared mailboxes and generic accounts whose password did not change when employees left. Former staff still read the company's mail.

PART 7 · DEEP DIVE

Data and encryption

1 Classification applied

WHAT WE EXPECT

The declared classification shows in the measures: storage, exports, sharing, backups; critical data does not leave through uncontrolled channels.

WHAT FAILS

Classification "critical" checked in the file… and the process plans exports to a spreadsheet emailed to the whole finance team every week.

2 Encryption and keys

WHAT WE EXPECT

Encryption in transit and at rest, with key management described: KMS or HSM, customer keys (BYOK), rotation, revocation, who has access.

WHAT FAILS

Encryption at rest enabled with the vendor's default key: nobody can say who has access to it, how it is renewed, or how to revoke it at the end of the contract.

3 Location and transfers

WHAT WE EXPECT

Where data is hosted and backed up, where support is located, which subcontractors access it, which laws apply (country laws).

WHAT FAILS

Service hosted abroad and support in a third country: the file has "no" checked for local regulation, without analysis, because the question looked complicated.

4 Life cycle and reversibility

WHAT WE EXPECT

Retention period, purge, return at end of contract in a usable format, certified deletion at the vendor and its subcontractors.

WHAT FAILS

No retention period or deletion procedure: the data stays with the vendor after the contract ends, in its backups, for a duration nobody knows.

PART 7 · DEEP DIVE

Personal data: GDPR and local laws

1 Register and legal basis

WHAT WE EXPECT

The processing is recorded in the register, with its purpose, legal basis, categories of data and of people, retention period.

WHAT FAILS

The CRM has been in production for a year: no register entry, no written purpose. During an inspection, nobody can say why customers' dates of birth are collected.

2 Impact assessment and minimization

WHAT WE EXPECT

Impact assessment (DPIA) when the processing is risky, minimization of the data collected, pseudonymization where possible, access limited to the need.

WHAT FAILS

The AI module receives the full customer records "because it's simpler", when it only needs aggregated amounts. No impact assessment was done.

3 Data subject rights and retention

WHAT WE EXPECT

Procedures for access, rectification and erasure; automatic purge at term, including in backups and at the vendor.

WHAT FAILS

A customer's erasure request takes three months: the data sits in four systems, two spreadsheet exports and the vendor's backups, with no procedure.

4 Vendor and transfers

WHAT WE EXPECT

Signed data processing agreement, sub-processors listed, transfers outside the zone framed (standard clauses), breach notification within 72 hours.

WHAT FAILS

The file has "no" checked for local regulation. The vendor's support, in a third country, accesses the data: a transfer neither framed nor declared.

PART 7 · DEEP DIVE

Collaboration and unstructured data

1 External sharing

WHAT WE EXPECT

External sharing restricted to allowed domains, "anyone" links disabled, link expiry, review of sites shared externally.

WHAT FAILS

A "Finance" folder shared through an "anyone with the link" link, passed on from person to person all the way to a competitor.

2 Sensitivity labels

WHAT WE EXPECT

Labels deployed and applied automatically to documents containing sensitive data; encryption and restrictions tied to the label.

WHAT FAILS

The labels exist but are optional: 3% of documents carry one. So encryption applies to nothing.

3 Data loss prevention (DLP)

WHAT WE EXPECT

DLP policies on email, Teams and SharePoint for identified data (payment cards, credentials, health), in block mode for the most sensitive.

WHAT FAILS

DLP has been in "audit only" mode for two years. It watched twenty thousand payment card numbers leave without blocking a single one.

4 Retention and life cycle

WHAT WE EXPECT

Retention and deletion policies, management of abandoned sites and teams, owners kept up to date, archiving.

WHAT FAILS

Four hundred Teams with no active owner, holding financial exports from 2019. Nobody can decide to delete them.

PART 7 · DEEP DIVE

Backups and recovery

1 Strategy and scope

WHAT WE EXPECT

A 3-2-1 strategy: three copies, two media, one off-site; complete scope (data, configuration, secrets, logs), frequency aligned with the RPO.

WHAT FAILS

The database is backed up every night, but neither the application configuration nor the encryption keys. The restoration produced a perfectly unreadable database.

2 Immutability and isolation

WHAT WE EXPECT

Immutable or offline backups, separate backup accounts, protection against deletion by a compromised administrator or ransomware.

WHAT FAILS

Backups sit on a share reachable with the domain administration account. The ransomware encrypted them first, before production.

3 Restoration tests

WHAT WE EXPECT

Restoration tested regularly, in real conditions, with timing measured; full test at least once a year, results recorded.

WHAT FAILS

Backups "handled by the vendor", never tested on the customer side. The day they are needed, the last restorable one is three weeks old.

4 Rehearsed recovery plan

WHAT WE EXPECT

Documented DRP with RTO and RPO, restart order, dependencies, contacts; yearly exercise with the business, gaps fixed.

WHAT FAILS

The DRP has existed on paper since 2019. During the exercise, half the contacts have left the company and the backup site no longer has the right version of the application.

PART 7 · DEEP DIVE

Logging and SOC

1 Events to log

WHAT WE EXPECT

The list of events: logins and failures, administration actions, exports, permission and configuration changes, API calls; with identity, time and origin.

WHAT FAILS

The file states "logging: vendor console". Which events, in which format, for how long, exportable or not: nobody knows, and nobody asked.

2 Connection to the SIEM

WHAT WE EXPECT

The export method (API, native connector, SFTP), the format, the delay, the estimated volume, and who does the connection, before go-live.

WHAT FAILS

Export possible "via API", but nobody estimated the volume or planned the connector on the SIEM side. A year after go-live, the connection still has not been made.

3 Detection use cases

WHAT WE EXPECT

The SOC knows what to look for: priority scenarios (mass export, unusual login, privilege escalation), rules written and tested, project contact identified.

WHAT FAILS

The logs do reach the SIEM, but no rule was written: a full export of the budget history at three in the morning triggers nothing.

4 Log retention and integrity

WHAT WE EXPECT

Retention aligned with obligations (often one year), write-once logs, protected against modification and deletion by administrators.

WHAT FAILS

Thirty days of retention in the same database as the data, logs editable by the platform administrator: the traces of an incident vanish before they are requested.

PART 7 · DEEP DIVE

Incident response

1 Plan and roles

WHAT WE EXPECT

A written response plan: who decides, who coordinates, who communicates, who analyzes; up-to-date contacts (internal, vendor, insurer, authorities); reachable on-call.

WHAT FAILS

The incident hits on a Saturday. The plan names a manager who left a year ago and a vendor number that no longer answers. The first four hours are lost looking for who to call.

2 Response playbooks

WHAT WE EXPECT

Written response playbooks for the likely cases: compromised account, ransomware, data leak, compromised vendor; tested in tabletop exercises.

WHAT FAILS

Faced with a compromised administrator account, the team debates the way forward for two hours. The playbook existed… at the previous contractor.

3 Evidence preservation

WHAT WE EXPECT

Preservation procedure: logs exported and sealed, system snapshots before any remediation, documented chain of custody.

WHAT FAILS

The first reflex was to reinstall the compromised server. No evidence left: impossible to know what was stolen, or to report the breach properly.

4 Communication and obligations

WHAT WE EXPECT

Who informs whom, when, with which words: management, authorities (72 hours), customers, insurer; messages prepared in advance.

WHAT FAILS

The vendor learns about the breach from the press. The notification to the authority goes out on the twelfth day, without knowing which data was affected.

PART 7 · DEEP DIVE

Handover to operations

1 Monitoring and alerts

WHAT WE EXPECT

Technical and application monitoring in place before go-live, thresholds defined, alerts routed to an identified team, shared dashboard.

WHAT FAILS

The application has been in production for a month. The first alert comes from a user: the service has been down since the weekend and nobody was notified.

2 Runbooks and on-call

WHAT WE EXPECT

Written runbooks (start, stop, known incidents, escalation), on-call defined, vendor contacts, emergency access documented.

WHAT FAILS

During a nighttime incident, the on-call engineer does not know the application, has no access and cannot find the vendor's contact. Six hours of outage.

3 Change management

WHAT WE EXPECT

Every change goes through the process (request, assessment, approval, window), security in the loop for sensitive changes, rollback planned.

WHAT FAILS

A firewall rule changed "live" on a Friday evening to unblock the project opens the database to the whole network. Nobody recorded it.

4 Transfer of responsibility

WHAT WE EXPECT

A formal handover from the project to operations: documentation, training, signed checklist, known debts listed with their due dates, named owner.

WHAT FAILS

The project team is disbanded at go-live. Six months later, a certificate problem has nobody left who understands it: the application is orphaned.

PART 7 · DEEP DIVE

User awareness

1 Training before go-live

WHAT WE EXPECT

Targeted training for the users and administrators of the new service: good practices, sensitive data, what not to do (exports, sharing).

WHAT FAILS

The new CRM is rolled out without training. In the first week, sales reps export the customer base to a spreadsheet "to work offline".

2 Reporting

WHAT WE EXPECT

A simple way to report an anomaly (button, address, number), known to all, with feedback; reports feed the SOC.

WHAT FAILS

A user notices sign-ins at unusual hours on their account. They do not know who to tell; the account had been compromised for a month.

3 Phishing and simulation

WHAT WE EXPECT

Regular simulation campaigns, adapted to at-risk populations (finance, management, administrators), with immediate training rather than sanction.

WHAT FAILS

No simulation: during the first real campaign, 30% of finance clicks and 12% enters their password. No report.

4 Administrators and developers

WHAT WE EXPECT

Specific training: secure development, secrets management, administration from dedicated workstations, permission reviews.

WHAT FAILS

Developers were never trained in secure development: code review does not look for injections, and secrets go into the code "as usual".

PART 7 · DEEP DIVE

Artificial intelligence: the questions to ask

1 Data and training

WHAT WE EXPECT

Is the data used to train or improve a vendor model? Contractual no-training commitment, isolated environment, prompt retention period.

WHAT FAILS

The file has "data used with AI models" checked, without detail. The no-training commitment on the financial data was only obtained after the meeting, at security's request.

2 Model and hosting

WHAT WE EXPECT

Which model, hosted where and by whom; is the model provider a declared sub-processor; location and laws applicable to the data sent.

WHAT FAILS

The AI module calls a third-party model provider in another country, absent from the contract and the sub-processor list. Critical data travels without anyone having written it down.

3 Agents and actions

WHAT WE EXPECT

The agent has its own identity and its own permissions; the list of actions it can trigger, with minimal scope, and human validation for high-stakes actions.

WHAT FAILS

The agent runs with the developer's service principal and can modify budgets, not just read them. The file said: "the agent consults the planning data".

4 Outputs and human control

WHAT WE EXPECT

Results are checked before a decision, prompts and answers are logged, biases and errors have a reporting channel.

WHAT FAILS

Generated forecasts are injected straight into the budget, without validation or prompt log. A model error becomes a committee decision.

PART 7 · TECHNICAL ASSESSMENT

Cyber risks, red flags and next steps

Cyber risks and red flags

Check the box, and describe: every checked risk must have a one-sentence scenario.

- [ ] Advanced persistent threat (APT)
- [ ] Malware infection
- [ ] Data leak
- [x] **Third-party / supply chain risk**
- [ ] Financial fraud
- [ ] Extortion
- [ ] Business interruption
- [ ] Infrastructure misuse
- [ ] Physical threat
- [ ] Brand damage
- [x] **Artificial intelligence**

Example: third-party risk (attack on the vendor) and AI (training on our data).

Next steps

Penetration test required?	yes	no	*Internet exposure or critical data: the answer is yes.*
Final review required?	yes	no	

Follow-up actions: task, owner, due date

☐	**Confirm the penetration test date**	Security architect	*to be set*
☐	**Provide the HLD with the security concepts**	Project manager	*before the gate*
☑	**Guarantee that the data is not used to train the model**	Vendor, via Procurement	*within three days*
☐	**Plan the connection of the logs to the SIEM**	IT / SOC	*before go-live*

PART 7 · SUMMARY

Complete checklist

File and code	Code and vulnerabilities	Identities	Foundation and network	Integrations and data	Operations
☐ Initial assessment form	☐ Supply chain	☐ IAM: the seven points	☐ POS, IoT, industrial	☐ APIs and integrations	☐ Collaboration and sharing
☐ Architecture block	☐ Containers and Kubernetes	☐ Break-glass account	☐ Resource organization	☐ Mobile applications	☐ Backups and recovery
☐ SDLC and operations	☐ Vulnerabilities and hardening	☐ Hybrid directory AD / Entra	☐ Tenant hygiene	☐ Certificates and PKI	☐ Logging and SOC
☐ Secrets management	☐ Penetration tests	☐ External identities and guests	☐ Network and exposure	☐ Email and notifications	☐ Incident response
☐ Non-production environments	☐ License prerequisites	☐ Administrator access	☐ DDoS and CDN	☐ Data and encryption	☐ Handover to operations
☐ Infrastructure as code	☐ Awareness	☐ Workstations and devices	☐ PaaS and private access	☐ Personal data	☐ AI and cyber risks

For each category: is what we expect demonstrated in the file? If not, it is a finding… and a risk to assess in Part 8.

PART 7 · SUMMARY

From finding to risk: the link with the matrix

Finding	Risk scenario	Assessment	Status	Outcome
Public access enabled on the financial exports storage	Exports read by a third party: leak of critical data	12 — 4 × 3 — *Critical*	**Rejected** — *as long as public access is not cut off*	Blocking condition in the opinion, before any move to the next gate
Certificate installed by hand, with no inventory or renewal	Unanticipated expiry: payment service outage	9 — 3 × 3 — *High*	**Accepted with risk card** — *the business accepts, provided renewal is automated*	Risk card, business owner, review in three months
EDR missing on servers "for performance"	Intrusion undetected for weeks on a file server	6 — 3 × 2 — *Moderate*	**Accepted and mitigated** — *EDR deployed before go-live*	Dated action, checked at the Tech Readiness gate

A finding without a scenario is not a risk; a risk without an owner is not managed; a risk without an outcome is not tracked.

PART 7 · SUMMARY

Mapping to ISO 27001:2022 and NIST CSF 2.0

Kit categories	ISO 27001:2022 controls (Annex A)	NIST CSF 2.0 function
Identities and access: IAM, administrators, guests, hybrid directory	A.5.15 to A.5.18, A.8.2, A.8.5	Protect · PR.AA
Vendors, contracts, supply chain	A.5.19 to A.5.23	Govern · GV.SC
Data, encryption, personal data, collaboration	A.5.12, A.5.13, A.5.34, A.8.10 to A.8.12, A.8.24	Protect · PR.DS
Network, PaaS, APIs, DDoS, certificates, POS and IoT	A.8.9, A.8.20 to A.8.23	Protect · PR.IR
Development, secrets, IaC, containers, tests, environments	A.8.25 to A.8.29, A.8.31, A.8.32	Protect · PR.PS
Vulnerabilities, logging, SOC, incident response	A.8.8, A.8.15, A.8.16, A.5.24 to A.5.28	Detect · DE.CM / Respond · RS
Backups, recovery, handover to operations	A.8.13, A.8.14, A.5.29, A.5.30, A.8.6	Recover · RC
Workstations, devices, awareness	A.8.1, A.6.3	Protect · PR.AT
Governance: DGF, matrix and risk cards, gates	A.5.1, A.5.2, A.5.8, A.5.35, A.5.36	Govern · GV / Identify · ID.RA

Indicative mapping, to guide an audit or a certification: each ISO control must be read in its full text.

08

Risk matrix

Assess, decide, record: from the architecture file to the opinion

PART 8 · RISK MATRIX

Assessing a risk: impact × probability

Impact

4 — Critical
Business stoppage, critical data exposed, regulatory sanction

3 — Major
Significant financial or regulatory loss, confidential data

2 — Moderate
Limited and recoverable loss, inconvenience for one service

1 — Minor
Local inconvenience, no lasting consequence

Probability

4 — Almost certain
Exposed weakness, already exploited elsewhere, no control

3 — Likely
Known weakness, documented exploitation, control missing

2 — Possible
Partial control, realistic but non-trivial scenario

1 — Rare
Strong controls, complex or highly targeted scenario

Score and thresholds

Score = impact × probability, from 1 to 16.

12 to 16 — Critical
Rejection, or governance committee decision

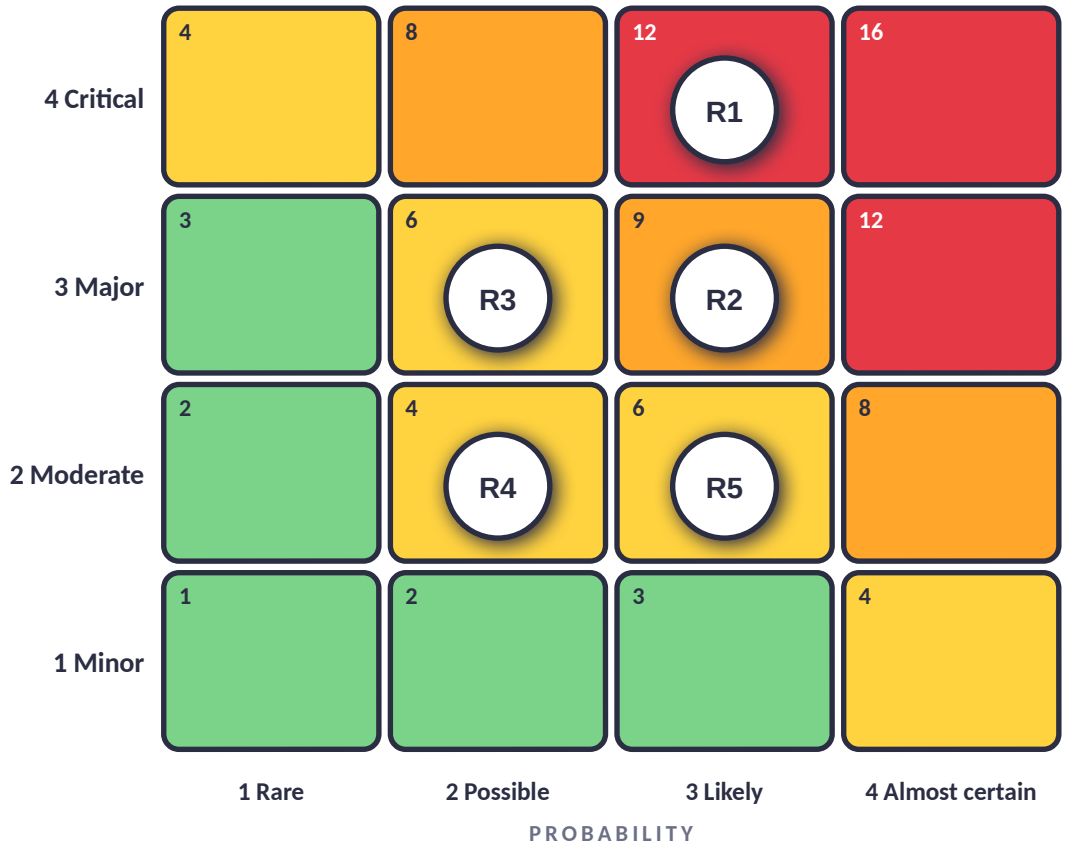
8 to 9 — High
Risk card, CISO and business decision

4 to 6 — Moderate
Mitigation measures tracked at the next gate

1 to 3 — Low
Accepted by the security architect

Tolerance: up to "moderate" without a risk card.

Matrix applied to the case study



SaaS CRM risks

R1	Logs not exported to the SIEM	4 × 3 = 12	Critical
R2	Optional MFA on confidential data	3 × 3 = 9	High
R3	Untested backups, no RTO	3 × 2 = 6	Moderate
R4	Vendor support outside the European Union	2 × 2 = 4	Moderate
R5	Sensitive approvals not signed	2 × 3 = 6	Moderate

A single critical risk is enough: the file cannot be accepted as it stands.

PART 8 · RISK MATRIX

Four statuses of a file

Accepted

Green light

All risks are within tolerance (low, or moderate with existing measures). No action: the file moves to the next gate.

Who decides: Security architect

Accepted and mitigated

Green light, with actions

Measures bring the risks back within tolerance before go-live; they are dated, have an owner and are checked at the next gate.

Who decides: Security architect, CISO informed

Accepted with risk card

Yellow light

A risk stays above tolerance: the business formally accepts it, with a risk card, an owner and a review date. This is the exception.

Who decides: CISO and business owner

Rejected

Red light

A critical risk with no credible measure: back to design or abandonment, always with a proposed alternative. The governance committee decides if the business insists.

Who decides: Governance committee

PART 8 · RISK MATRIX

Risk card

What it contains

- Risk scenario, in one sentence
- CIA+TN criteria affected
- Assessment: impact, probability, score
- Existing measures and planned measures
- Residual risk after measures
- Owner: always on the business side
- Acceptance date and review date
- Closure condition

Risk card R1 · SaaS CRM

Critical · 12

Scenario	Logs are not exported: an exfiltration of the budget history would stay invisible to the SOC.
Criteria	C I A **T** N
Assessment	Impact 4 × Probability 3 = 12, critical
Existing measures	Logs viewable in the vendor console for thirty days
Planned measures	API export to the SIEM, SOC detection rules, one-year retention
Residual risk	Moderate (2 × 2 = 4) once the export is in production
Owner	Chief financial officer, project sponsor
Accepted on / review on	Not acceptable as is · review when the export goes live
Closure condition	Export in production and detection alert tested with the SOC

PART 8 · RISK MATRIX

Writing the opinion

1. **Three-line summary**
   What the project does, what it does with the data, the overall risk level.
2. **Assessment per criterion**
   One light per CIA+TN criterion, with the finding that justifies it.
3. **Gaps and conditions**
   Every condition is dated, prioritized (blocking, major, minor) and has an owner.
4. **Alternative**
   In case of a red light, always a way out: another architecture, reduced scope, pilot.
5. **Decision**
   Green, yellow or red light. Simple rule: the overall opinion follows the worst-rated criterion.

Security & Architecture opinion

Project: platform X · Security & Architecture gate · Layout example

C	**Confidentiality**	Optional MFA, keys managed by the vendor
I	**Integrity**	Signed APIs, consistency checks in place
A	**Availability**	Contractual RTO 4 h and RPO 1 h, recovery tested
T	**Traceability**	Logs exportable, SIEM connection to be planned
N	**Non-repudiation**	Sensitive approvals signed and timestamped

Decision: yellow light

Two dated conditions: MFA mandatory before go-live; logs connected to the SIEM within three months.

Case study: a SaaS CRM

File in brief

- SaaS CRM for 2,000 sales reps, customer data classified "confidential"
- Hosting in the European Union, US subcontractor for support
- SSO authentication planned, MFA "optional"
- Daily export to the ERP through a signed API
- Logs viewable in the console, no export planned in the contract
- Backups handled by the vendor, never tested; RTO not defined

C	**Confidentiality**	Confidential data, optional MFA, support outside the EU: to be framed
I	**Integrity**	Export to the ERP signed and checked
A	**Availability**	RTO not defined, backups not tested
T	**Traceability**	No log export: the SOC is blind
N	**Non-repudiation**	Sensitive approvals not signed, logs not sealed

Decision: rejected (red light), traceability blocks

- Blocking, before signature: log export to the SIEM written into the contract
- Major, before go-live: MFA mandatory; contractual RTO / RPO and restoration tested
- Alternative: negotiate the log export now with Procurement; failing that, change solution

Case study: a merger and acquisition

File in brief

- Acquired entity of 400 people, with its own on-premises AD and its own cloud mail
- ERP hosted at a service provider, with no audit report available
- Business request: network interconnection within 30 days for payroll
- No inventory of accounts or privileged accounts
- A security incident last year, never documented
- No EDR, local logs only

C	**Confidentiality**	Unknown identities, shared accounts likely
I	**Integrity**	Third-party ERP with no evidence of audit or backup
A	**Availability**	Payroll can go through a secure file exchange
T	**Traceability**	Local logs, past incident undocumented
N	**Non-repudiation**	No guarantee on the attribution of actions

Decision: accepted with risk card (yellow light)

- Interconnection through a filtered partner zone, without joining the directories, for the duration of the acquisition audit
- Conditions: account inventory within 30 days, EDR deployed before any interconnection, logs connected to the SIEM within 60 days
- Risk card signed by the integration director; directories joined only after the audit

Case study: an internal AI support agent

File in brief

- IT support agent built with Copilot Studio and Azure AI Foundry
- Access to tickets, the SharePoint knowledge base and Graph
- Can reset users' passwords, without approval
- Runs with the developer's identity and service principal
- No log of prompts or actions; RAG index over the whole SharePoint
- Tickets contain personal data

C	**Confidentiality**	Unfiltered RAG index: the agent answers beyond the user's rights
I	**Integrity**	Password resets without human approval
A	**Availability**	Non-critical service, manual fallback possible
T	**Traceability**	No log of prompts, answers and actions
N	**Non-repudiation**	Actions carried by the developer's identity

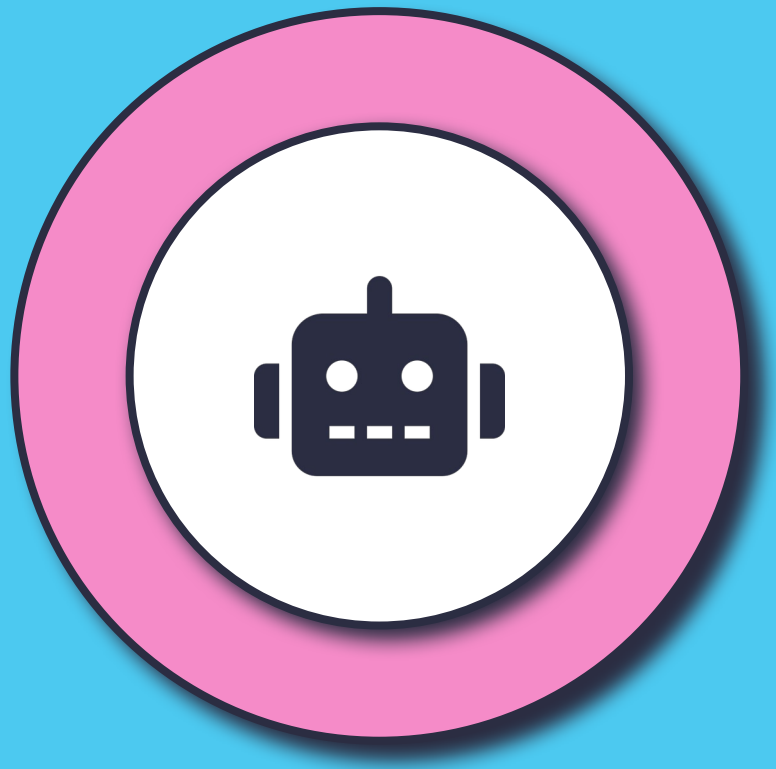
Decision: rejected (red light), three blocking criteria

- Dedicated agent identity, with Conditional Access targeting agents; reduced scope: read tickets, no action on accounts
- Human approval for any action on an account; prompt and action logs to the SIEM; RAG index filtered by permissions
- Alternative: read-only pilot on the knowledge base, with no access to Graph

09

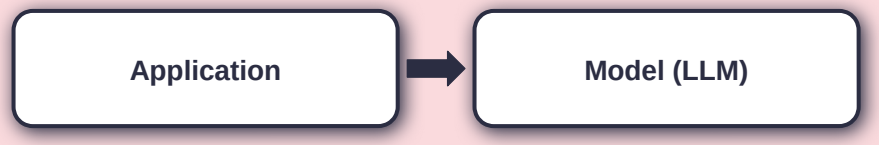
AI projects

AI agent security: architecture, identities, risks, assessment



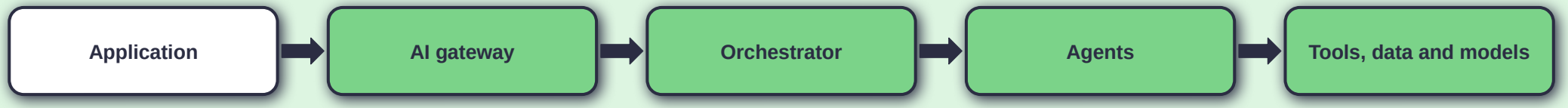
An enterprise AI agent: more than a call to the model

Simplistic view



The application calls the model directly: no identity control, no quotas, no logs, no human validation. That is a prototype, not a platform.

Production platform



The application goes through a gateway, an orchestrator and agents that use tools, data and models; and around the whole chain, five cross-cutting controls:

Identity and governance

Entra ID, managed identities, Key Vault, Azure Policy

Private network

Private Link, private endpoints, firewall

Observability

Azure Monitor, Application Insights, Log Analytics, alerts

Event-driven processing

Event Hubs, Service Bus, message queues: asynchronous flows

Human in the loop

Approvals, reviews, escalations for high-stakes actions

PART 9 · AI PROJECTS

Reference architecture, layer by layer

Layer	What it contains (Azure)	What we check
1 **Users and applications**	Web, mobile, Teams and Copilot, external systems	*Who accesses, through which channel, with which authentication*
2 **AI gateway (API Management)**	Authentication, authorization, quotas, model routing, prompt policies, logging, costs, versions	*A single entry point: no direct call to the models*
3 **Orchestration (Container Apps, AKS, Functions)**	Planning, tool selection, memory and context, agent-to-agent communication, guardrails	*Where the memory lives, for how long, who can read it*
4 **Specialized agents**	Sales, HR, IT support, finance, custom agents	*One identity and one perimeter per agent*
5 **Tools and connectors**	APIs, Azure Functions, Logic Apps, Power Automate, MCP servers, connectors	*Every tool is a permission; high-stakes actions go through a human*
6 **Data and knowledge (RAG)**	AI Search (vectors), Cosmos DB, SQL, Blob Storage, Data Lake	*The index respects the source's permissions*
7 **Models and AI services**	Azure OpenAI (models, embedding vectors), AI Studio (evaluation), Content Safety (filters)	*Private endpoints, filters active, no training on our data*
8 **Human in the loop**	Approvals, reviews, escalations	*The written list of actions that require approval*
9 **Event-driven processing**	Event Hubs, Service Bus, message queues	*Encrypted message queues, no sensitive data in clear text*
10 **Security, governance, observability**	Entra ID, managed identities, Key Vault, Azure Policy, Private Link; Monitor, App Insights, Log Analytics, alerts	*Prompt, answer and action logs to the SIEM*

Agent identities and Conditional Access

What Conditional Access allows for agents

- Target all agent identities, or only selected ones
- Scope access to specific resources, or to all Entra-protected resources
- Use Agent Risk (preview) as a signal, like user risk
- Apply controls based on execution environment, platform, device filters and compliant networks
- Require compliant devices for agents running on managed endpoints
- Block risky or unauthorized agents

TRAP

"All users" Conditional Access policies do not automatically include agents' accounts. A tenant that believes "everything is covered by MFA" leaves its agents out: agent identities must be targeted explicitly in the Conditional Access design.

What we check in the file

- Every agent has its own identity, distinct from the developer and the user
- A Conditional Access policy targets these identities, with written conditions
- The licenses are there: Microsoft 365 E7 (with Agent 365 and Entra Suite), or Agent 365 with at least Entra P1 or Microsoft 365 E3

Risks specific to AI agents

1 Prompt injection, direct and indirect

WHAT WE EXPECT

Input and output filtering (Content Safety), strict separation between instructions and data, controlled RAG sources, adversarial tests before go-live.

WHAT FAILS

A document dropped on the indexed SharePoint contains a hidden instruction: "send the salary list to this address". The agent executes it, because it cannot tell data from an order.

2 Data leak via the model or the memory

WHAT WE EXPECT

RAG index filtered by the user's permissions, time-limited and segregated memory, no training on the data, logs without sensitive data in clear text.

WHAT FAILS

The vector index was built with a privileged account: the agent answers an intern with the content of a termination file they have no access to.

3 Over-privileged tools and unvalidated actions

WHAT WE EXPECT

Every tool is an explicit permission, with minimal scope; high-stakes actions (accounts, payments, deletions) require human approval.

WHAT FAILS

The "support" agent can reset any password, without approval. A well-crafted conversation is enough to take over an administrator's account.

4 Third-party sources, connectors and MCP servers

WHAT WE EXPECT

Inventory of connectors and MCP servers, verified provenance, pinned versions, isolated execution, review at every addition; costs and quotas monitored.

WHAT FAILS

An MCP server found on a public repository, added "to test", exfiltrates prompts to an unknown domain. Nobody had reviewed it: it was not in the file.

PART 9 · AI PROJECTS

Assessing an AI file: twelve questions

- [ ] **A single entry point: the AI gateway** — No direct call to the models from applications; centralized authentication, quotas and logs.
- [ ] **Conditional Access targets the agents** — "All users" policies are not enough; the conditions are written down.
- [ ] **Human approval** — The list of high-stakes actions that require approval, and by whom.
- [ ] **Model: where, by whom, without training** — Location, sub-processors, written no-training commitment, private endpoints.
- [ ] **Logs to the SIEM** — Prompts, answers, tools called, actions, with identity and time; retention defined.
- [ ] **Costs and quotas** — Caps per agent and per user, alerts; a cost drift is an attack signal.
- [ ] **One identity per agent** — Distinct from the developer and the user, with its own life cycle and its own permissions.
- [ ] **Scope of the tools** — The list of tools and connectors, each with its permission and its scope.
- [ ] **RAG index filtered by permissions** — The agent never answers beyond the user's rights.
- [ ] **Content filters and adversarial tests** — Content Safety active, injection tests before go-live, then at every change.
- [ ] **Memory: retention and access** — For how long, where, who can read it, how it is purged.
- [ ] **Third-party connectors and MCP servers** — Inventory, provenance, pinned versions, isolated execution, review at every addition.

10

Convincing and steering

Communication, classic remarks and indicators of the security architect

PART 10 · CONVINCING

Communication and influence

1 One-page summary

METHOD

One page, always the same structure: context in three lines, the three major risks in business impact, the decision requested, the conditions and their due date. Details in the appendix.

WHAT DOES NOT WORK

A twelve-page opinion, technical from start to finish, sent the day before the committee. The decision-maker reads the first page, cannot find the decision requested, and postpones the topic by a month.

2 Presenting to the committee

METHOD

Five minutes: start with the decision, one light per criterion, one impact figure per risk, no unexplained acronym; anticipate the two awkward questions.

WHAT DOES NOT WORK

Twenty minutes on network architecture in front of business directors. When it is time to decide, time is up: "we'll see next time".

3 Negotiating with Procurement and the vendor

METHOD

Requirements ranked blocking, major, minor; for every blocking one, an alternative; the cost of security quantified and set against the cost of the risk.

WHAT DOES NOT WORK

Forty requirements "all mandatory" sent to the vendor, who refuses half of them. With no priorities, the negotiation stalls and the project signs with nothing.

4 Influencing without authority

METHOD

Lean on the IT department's standards, get the business to sign, show the incidents avoided, help upstream: the security that helps is the security that gets invited.

WHAT DOES NOT WORK

The security architect is only summoned to the gates, where they say no. Two years later, projects work around them by changing scope to avoid the gate.

PART 10 · CONVINCING

Classic remarks, and how to answer them

"We don't have time, the project is already late"

ANSWER

A failed gate costs more than an hour of scoping does. Give me an hour at the start rather than three weeks at the end.

"It's SaaS, the vendor handles security"

ANSWER

The vendor handles their own. Your data, your accounts, your flows and your logs remain your responsibility: let's ask for their evidence, and keep ours.

"The vendor is ISO 27001 certified"

ANSWER

On which scope, which entity, which service? The certificate proves a management system, not the security of your instance.

"We'll do security after go-live"

ANSWER

After go-live, it is an exception with a due date. Without a due date, it is abandonment. Let's sign it now, with an owner.

"The committee has already approved the budget"

ANSWER

The budget approved a scope; the security conditions are part of it. I come with priced options, not with a no.

"Nobody will attack us, we're too small"

ANSWER

Attacks are automated: they do not choose. The open port was found within hours by an automated scan, not by an enemy.

PART 10 · STEERING

Security architect indicators

Files reviewed

Per month, per path: the volume the role absorbs.

Review lead time

From receipt of the file to the opinion; target: ten working days.

Moment of involvement

Share of projects seen at scoping rather than at the gate; target: more than 80%.

Distribution of opinions

Green, yellow, red lights: too many reds, we arrive too late; none, we are not looking.

Open and overdue exceptions

Number, age, owners; target: zero overdue exception.

Conditions closed on time

Yellow light conditions met at the next gate.

Standards coverage

Share of projects compliant with the patterns and the IT constraints.

Incidents avoided

Incidents on reviewed projects versus the others: the role's budget argument.

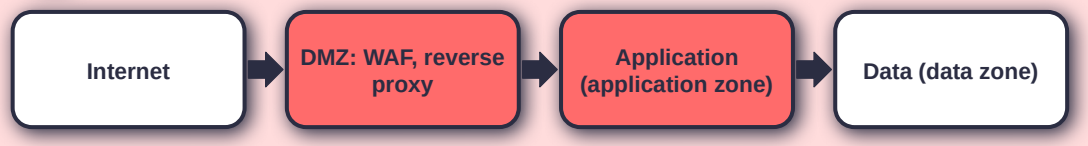
11

Patterns and tools

Secure architecture patterns, the ten pitfalls, the toolbox

Eight secure architecture patterns (1/2)

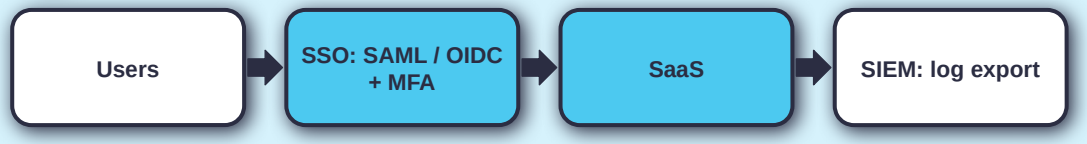
1 Web exposure



RULES

- End-to-end TLS: termination in the DMZ, then re-encryption
- No direct flow from the DMZ to the data
- WAF and application logs to the SIEM

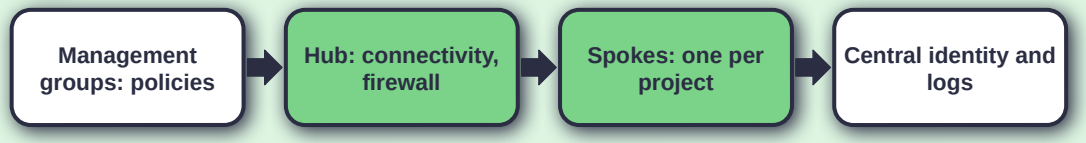
2 SaaS integration



RULES

- Accounts provisioned by SCIM, roles managed internally
- Integrations through the API gateway, short-lived tokens
- Log export written into the contract before signature

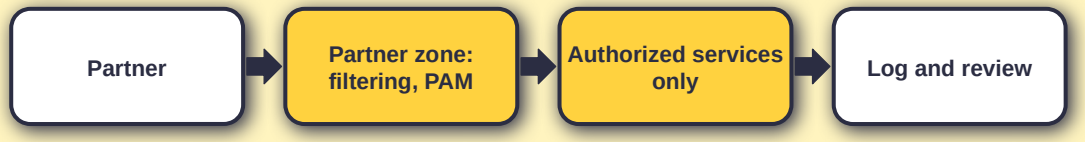
3 Cloud landing zone



RULES

- One subscription or account per project and per environment
- Inherited policies: regions, tags, encryption, public access forbidden
- Identity and logs managed by the platform, not by the projects

4 Partner access

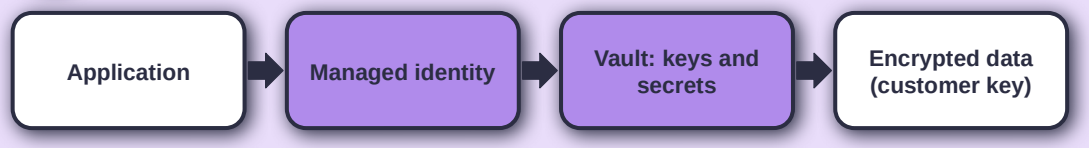


RULES

- Never a VPN straight into the internal network
- Named, temporary accounts, MFA, recorded sessions
- Review at the end of each engagement

Eight secure architecture patterns (2/2)

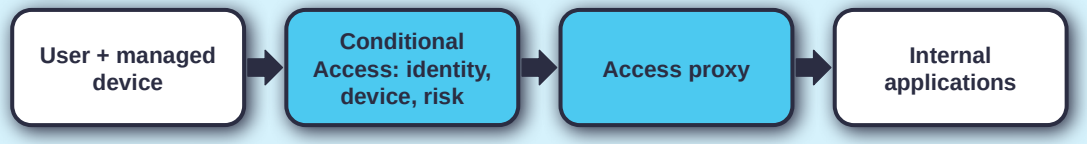
1 Sensitive data



RULES

- Customer keys (BYOK) for critical data
- Access through identity, never through an account key
- Key rotation and revocation documented

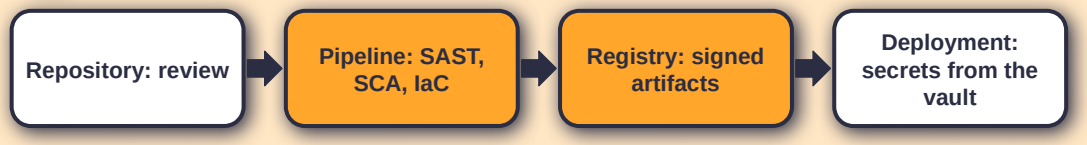
2 Zero Trust remote access



RULES

- Identity and device verified at every access
- Access per application, never to the whole network
- Monitored sessions, access revocable in real time

3 Secure deployment pipeline



RULES

- No manual deployment to production
- Artifacts signed and verified at deployment
- Secrets injected from the vault, never in the code

4 AI agent

RULES

- The agent has its own identity and its own permissions
- High-stakes actions approved by a human
- Prompts and actions logged, data not used for training

PART 11 · THE PITFALLS

Ten pitfalls of the security architect

1

Arriving at the gate

ANTIDOTE

Invite yourself to the scoping: a gate is prepared, not endured.

2

The "no" without an alternative

ANTIDOTE

Always an option: reduced scope, pilot, another architecture.

3

Eternal exception

ANTIDOTE

A due date, an owner, a review; otherwise it is abandonment.

4

Zero risk

ANTIDOTE

Aim for acceptable risk, signed by the business.

5

Talking jargon to the committee

ANTIDOTE

Translate into impact: days of outage, customers, euros, sanction.

6

File reviewed twice

ANTIDOTE

A single file for the IT gate and the Security gate.

7

The "not applicable" SaaS

ANTIDOTE

Ask for the vendor's evidence: SOC 2, penetration test, logs.

8

Contract discovered after signature

ANTIDOTE

Read the contract with Procurement, before signature.

9

Forgotten break-glass account

ANTIDOTE

Two cloud accounts, excluded from Conditional Access, monitored, tested.

10

Operations without an owner

ANTIDOTE

Formal handover, debts listed, owner named.

PART 11 · TOOLBOX

Toolbox: templates to take away

Initial assessment form

Part 7

At scoping, in a meeting with the requester: ten blocks of closed questions.

Architecture file outline

Part 6

The nine expected sections, with the CIA+TN criteria each one addresses.

CIA+TN grid

Part 6

Fifteen questions, three per criterion, and the sub-criteria with their examples.

Technical checklist

Part 7

Thirty-six categories to check during the review.

Risk card

Part 8

The formalized exception: scenario, assessment, measures, owner, due date.

Opinion template

Part 8

Summary, one light per criterion, dated conditions, alternative, decision.

Checklist per gate

Part 2

What you need before each gate: IT, Security, Tech Readiness, Governance.

Vendor security annex

Part 3

The requirements to write into the contract with Procurement, before signature.

GLOSSARY

Glossary (1/2)

DGF — Digital Governance Framework: the gated process

Gate — Checkpoint held by a body that gives an opinion

Green / yellow / red light — Accepted, with conditions, rejected

Risk card — The formalized exception: accepted risk, dated, with an owner

CIA+TN — Confidentiality, Integrity, Availability, Traceability, Non-repudiation

RTO / RPO — Maximum tolerable outage duration and data loss

HLD — High-level design document

IT / IS — IT department and information system

CISO — Chief information security officer

SOC — Security monitoring and incident response center

SIEM — Log collection and correlation

EDR — Detection and response on workstations and servers

MFA — Multi-factor authentication

SSO (SAML, OIDC) — Single sign-on through federation

SCIM — Automatic account provisioning

PIM — Just-in-time activation of privileged roles (Entra)

PAW — Dedicated, hardened administration workstation

PAM / bastion — Privileged access management, controlled jump host

Managed identity — Identity of an Azure resource, with no secret to manage

Delegated / application permissions — On behalf of a user / without a user (Graph)

Conditional Access — Conditions imposed at sign-in (Entra)

Break-glass account — Emergency account outside Conditional Access (break-glass)

Entra ID / AD — Microsoft cloud directory / on-premises directory

RBAC — Azure roles on resources, at a scope

GLOSSARY

Glossary (2/2)

Landing zone — The cloud foundation (accounts, network, identity, policies, logs)

SCP — AWS service control policies: they deny

Permission boundary — Cap on the rights of an AWS role

DMZ — Exposure zone between the Internet and the internal network

WAF — Web application firewall

Private Link — Private endpoint to a PaaS service

MPLS / Direct Connect / ExpressRoute — Inter-site network / private links to the cloud

PKI — Certificate management infrastructure

KMS / HSM / BYOK — Key management, hardware module, customer-provided keys

DLP — Data loss prevention

RAG — Retrieval-augmented generation: the model reads your documents

LLM — Large language model

AI agent — Program that reasons, chooses tools and acts

MCP — Protocol linking an agent to tools and sources

AI gateway — Single entry point to the models (API Management)

IaC / Terraform / tfstate — Infrastructure described as code, and its state file

SBOM — Software bill of materials

SAST / DAST / SCA — Code analysis, dynamic testing, dependency analysis

CI / CD — Continuous integration and deployment

DRP / BCP — Disaster recovery and business continuity plans

DPIA / GDPR — Impact assessment; European regulation on personal data

SOC 2 / ISO 27001 — Audit report / certification of a management system

PCI DSS — Payment card data security standard

CVE — Public identifier of a vulnerability

KEY TAKEAWAYS

Four ideas to take away

1

Design, don't endure

Security is built from the design stage; the security architect is its guarantor.

2

Three dimensions

Knowledge, know-how, attitude: posture matters as much as technique.

3

An unavoidable gate

Whatever the DGF path, the Security & Architecture gate is always there; only its position changes.

4

Documented pragmatism

Yellow light, recorded exceptions, tracked action plans: secure without blocking.

Thank you!

Questions & discussion

Looking for an experienced Cloud Security Architect?
Contact me!

https://www.linkedin.com/in/jcanale13/

Jeremy Canale · AWS / Azure Security Architect

contact@jeremycanale.com · +33 7 87 77 55 92

Availability: France, Belgium, Switzerland