

MÉTHODE & CONCEPTION

LIVRE BLANC MÉTHODOLOGIQUE

Concevoir une architecture de système d'information

Du cadrage métier à la trajectoire de mise en œuvre — méthode, normes et démarche

Comprendre ce qu'est une architecture SI, par quoi commencer, comment itérer, et sur quels cadres, normes et méthodes s'appuyer.

SOMMAIRE

- Pourquoi structurer son SI est stratégique
- Le périmètre de ce document
- Profils de lecture
- **Partie 1 — Qu'est-ce qu'une architecture de système d'information ?**
 - 1.1 Définition et finalité
 - 1.2 Architecture d'entreprise, architecture SI, architecture logicielle
 - 1.3 Les domaines d'architecture
 - 1.4 La sécurité, une préoccupation transverse
 - 1.5 Décrire une architecture : parties prenantes, points de vue, vues
 - 1.6 Ce qui relève de l'architecture SI, ce qui n'en relève pas
- **Partie 2 — Les cadres et les normes de référence**
 - 2.1 Les cadres d'architecture d'entreprise
 - 2.2 Les normes mobilisables
 - 2.3 Quel cadre pour quelle échelle
- **Partie 3 — Modéliser : langages et diagrammes de flux**
 - 3.1 « Diagramme de flux » : un terme, plusieurs réalités
 - 3.2 Les langages de modélisation
 - 3.3 Les outils
 - 3.4 Où se place chaque modèle
- **Partie 4 — La démarche : par quoi on commence, par quoi on finit**
 - 4.1 Le principe fondateur : le métier avant la technologie
 - 4.2 Deux activités transverses à tout le cycle
 - 4.3 Le cycle en phases
 - 4.4 L'analyse d'écart, moteur de la trajectoire

- 4.5 Détail par phase
- 4.6 Le début et la fin, explicitement

→ **Partie 5 — Itérer et gouverner**

- 5.1 Une cible et une trajectoire, pas un livrable figé
- 5.2 Le raffinement par paliers et la gouvernance d'architecture
- 5.3 Le rôle de l'architecte et le référentiel
- 5.4 L'architecture évolutive
- 5.5 Les trois échelles d'itération

→ **Partie 6 — Choisir les outils et les logiciels**

- 6.1 La méthode de décision rationnelle
- 6.2 Évaluer un logiciel open source : QSOS
- 6.3 Tracer la décision : l'ADR
- Arbre de décision
- Matrice cadre × contexte
- Scénarios types

→ **À propos de l'auteur**

Préface

Quelques années de missions de conseil et de mise en œuvre suffisent pour observer un schéma récurrent : l'expression « architecture SI » recouvre, selon les interlocuteurs, des réalités très différentes. Pour les uns, c'est un schéma de serveurs ; pour d'autres, la liste des logiciels installés ; pour d'autres encore, le travail du développeur. Cette confusion n'est pas anodine. Elle conduit des organisations à engager des budgets considérables sur des fondations mal posées.

Trois malentendus reviennent systématiquement. Le premier consiste à partir de la solution — un outil, une technologie à la mode, aujourd'hui « une intelligence artificielle » — avant d'avoir formulé le besoin métier qu'elle est censée servir. Le deuxième confond l'architecture du système d'information, qui orchestre l'ensemble des applications d'une organisation, avec l'architecture logicielle, qui décrit la structure interne d'une seule application. Le troisième mélange la conception, qui définit la cible, et l'exploitation, qui fait tourner le système au quotidien.

Ce document a été écrit pour dissiper ces malentendus. Il ne prétend pas remplacer les ouvrages de référence ni les normes qu'il cite : il les met en perspective et propose une démarche praticable, en distinguant clairement ce qui relève de l'architecture SI de ce qui n'en relève pas. Il s'adresse autant au décideur qui veut comprendre ce qu'il finance qu'au praticien qui veut structurer son travail.

Le fil conducteur tient en une conviction, posée d'emblée et défendue tout au long du document.

***Une architecture ne commence jamais
par un choix de technologie. Elle
commence par un besoin.***

Mattieu Pottier — MP-i · Mattieu Pottier Indépendant

Introduction

Concevoir un système d'information revient à organiser la manière dont une organisation produit, échange et exploite son information pour servir ses objectifs. L'architecture SI est la discipline qui rend cette organisation cohérente, durable et maîtrisée — plutôt que subie.

Pourquoi structurer son SI est stratégique

Faute d'architecture, un système d'information se construit par accréation : on ajoute une application ici, une intégration là, un correctif ailleurs. Au bout de quelques années, plus personne ne sait précisément qui parle à qui, quelle donnée fait référence, ni pourquoi tel choix a été fait. Les conséquences sont mesurables : dette technique, dépendances non maîtrisées, doublons fonctionnels, coûts d'exploitation qui dérivent.

L'architecture SI inverse cette logique. Elle aligne le système d'information sur la stratégie métier, arbitre les choix avant qu'ils ne deviennent irréversibles, et rend les décisions traçables. Un système mal architecturé n'est pas seulement plus difficile à faire évoluer : il coûte durablement plus cher à exploiter.

Une réserve, toutefois, vaut avertissement. Selon les analyses du cabinet Gartner, une part importante des livrables d'architecture — de l'ordre du tiers — n'est jamais réutilisée par le reste de l'organisation. L'architecture n'a de valeur que si elle est actionnable et reliée aux décisions réelles. Un document d'architecture qui ne sert à personne n'est pas de l'architecture : c'est du formalisme.

Le périmètre de ce document

Ce livre blanc couvre les définitions et le périmètre de l'architecture SI, les cadres et normes de référence, les langages de modélisation, la démarche de conception phase par phase, les mécanismes d'itération et de gouvernance, ainsi que la méthode de sélection des outils et des logiciels.

Il ne constitue pas un manuel exhaustif d'un cadre particulier, n'impose aucune technologie, et n'aborde ni la gestion de projet ni la gestion des services au quotidien, qui relèvent d'autres disciplines. Ces frontières sont assumées : elles font partie de la clarification que ce document cherche à apporter.

Profils de lecture

COMMENT LIRE CE DOCUMENT SELON VOTRE PROFIL		
PROFIL	PARTIES PRIORITAIRES	OBJECTIF
Décideur, DSI	Introduction, Parties 1 et 2, Guide décisionnel, Tendances	Comprendre le périmètre et cadrer une commande
Architecte, développeur	Parties 1 à 6	Disposer d'une démarche et d'un référentiel de normes
Chef de projet, product owner	Introduction, Parties 4 et 5, Guide décisionnel	Situer l'architecture dans le cycle projet

Partie 1 — Qu'est-ce qu'une architecture de système d'information ?

Avant toute démarche, il faut s'entendre sur les termes. Cette partie pose les définitions, distingue les niveaux d'architecture que l'on confond souvent, et présente la manière dont une architecture se décrit.

1.1 Définition et finalité

Architecture SI — Organisation d'ensemble qui relie la stratégie et les processus métier d'une organisation aux applications, aux données et à l'infrastructure qui les supportent. Elle décrit comment les différentes briques s'agencent et communiquent, afin que le système d'information serve les objectifs de l'organisation de façon cohérente et durable.

La finalité de l'architecture SI n'est pas de produire de beaux schémas, mais de prendre et de tracer de bonnes décisions structurantes : quels processus outiller, avec quelles applications, autour de quelles données, sur quelle infrastructure, et selon quels principes. Tout le reste en découle.

1.2 Architecture d'entreprise, architecture SI, architecture logicielle

La plupart des cadres invoqués dans ce domaine — TOGAF, Zachman — sont des cadres d'**architecture d'entreprise** : un périmètre plus large que le seul système d'information, puisqu'ils incluent la stratégie et l'organisation. L'architecture SI en est un sous-ensemble, et l'architecture logicielle, un cran en dessous encore, ne concerne que la structure interne d'une application. Confondre ces trois niveaux est la première source de malentendus.

TROIS NIVEAUX À NE PAS CONFONDRE

CRITÈRE	ARCHITECTURE D'ENTREPRISE	ARCHITECTURE SI	ARCHITECTURE LOGICIELLE

Périmètre	Stratégie, métier et SI	Ensemble des applications et données	Structure interne d'une application
Question centrale	Comment l'organisation crée de la valeur	Comment les systèmes se coordonnent	Comment une application est construite
Lectorat	Direction, stratégie	DSI, architectes	Développeurs, responsables techniques
Cadre type	TOGAF, Zachman	Urbanisation, TOGAF	Clean Architecture, patterns

1.3 Les domaines d'architecture

L'architecture SI se décline classiquement en plusieurs domaines. Le référentiel TOGAF en distingue quatre, désignés par l'acronyme BDAT : Business, Data, Application, Technology.



Architecture métier

Processus, fonctions, acteurs, organisation.
Décrit ce que fait l'entreprise, indépendamment de tout outillage.



Architecture des données

Modèles de données, référentiels, flux et gouvernance de l'information. Décrit ce que l'organisation connaît et manipule.



Architecture applicative

Applications et services, leurs responsabilités et leurs échanges. Décrit avec quoi les processus sont outillés.



Architecture technique

Infrastructure, réseau, hébergement, déploiement. Décrit où et comment l'ensemble s'exécute.

Ce découpage n'est pas universel : il varie selon le référentiel. L'urbanisation française distingue traditionnellement une couche *fonctionnelle* de la couche *applicative*, et traite la donnée de façon transverse. Le langage ArchiMate, lui, ne retient que trois couches de cœur — métier, applicative et technologique. Retenir un modèle revient donc moins à mémoriser un nombre de couches qu'à comprendre la logique de séparation : ce que fait l'organisation, ce qu'elle connaît, avec quoi elle l'outil, et sur quoi cela tourne. Dans la suite de ce document, les termes « domaine » et « couche » sont employés indifféremment.

1.4 La sécurité, une préoccupation transverse

Il serait tentant de ranger la sécurité dans la seule couche technique. C'est une erreur. La sécurité traverse les quatre domaines : elle concerne les droits métier, la confidentialité des données, le cloisonnement des applications et le durcissement de l'infrastructure. Les versions récentes des cadres de référence l'ont reconnu en élevant la sécurité au rang de préoccupation de premier plan.

▲ Point d'attention

La sécurité ne se rajoute pas à la fin : elle se conçoit dès le départ, à chaque couche. C'est le principe de sécurité par conception (*security by design*), repris plus loin dans la démarche.

1.5 Décrire une architecture : parties prenantes, points de vue, vues

La norme ISO/IEC/IEEE 42010:2022 définit ce qu'est une description d'architecture. Sa logique est essentielle à comprendre, car elle structure tout livrable sérieux.

LA LOGIQUE D'ISO/IEC/IEEE 42010

- › Des parties prenantes — direction, métier, exploitants, utilisateurs — portent des préoccupations.
- › À chaque préoccupation correspond un point de vue : une manière convenue de représenter l'architecture.
- › Un point de vue appliqué à l'architecture produit une vue : un diagramme ou un modèle concret.

- › Une description d'architecture complète n'est pas un schéma unique, mais un ensemble cohérent de vues.

Autrement dit, il n'existe pas de représentation unique et universelle d'une architecture : il existe autant de vues que de préoccupations à adresser. C'est pourquoi un projet d'architecture produit plusieurs diagrammes, et non un seul « grand schéma ».

1.6 Ce qui relève de l'architecture SI, ce qui n'en relève pas

RELÈVE DE L'ARCHITECTURE SI

La cartographie des processus et des capacités métier

Le découpage en applications et services, et leurs échanges

Le modèle de données et les référentiels

Les choix d'infrastructure structurants et la sécurité d'ensemble

Les principes directeurs et les décisions tracées

N'EN RELÈVE PAS

Le développement et le code des applications

L'exploitation et le maintien en condition opérationnelle

La gestion de projet et la planification des ressources

Le design d'interface détaillé

Le choix d'un composant ou d'une bibliothèque précise

◆ À retenir

L'architecture SI orchestre les applications entre elles à l'échelle de l'organisation. Elle s'arrête au niveau « on a besoin d'une interface programmable » ou « d'un référentiel unique de la donnée » — pas « on utilise tel framework ».

Partie 2 — Les cadres et les normes de référence

On ne conçoit pas une architecture à partir d'une page blanche : des cadres méthodologiques et des normes internationales structurent la démarche. Encore faut-il savoir lequel mobiliser, et à quelle échelle.

2.1 Les cadres d'architecture d'entreprise

STANDARD MONDIAL

TOGAF

10e éd., 2022 • The Open Group

Le cadre d'architecture d'entreprise le plus répandu. Son cœur, l'ADM, décrit un cycle de conception en phases. Conçu pour les grandes organisations : à adapter, pas à appliquer intégralement.

ADM

EA

ÉCOLE FRANÇAISE

Urbanisation du SI

Années 2000 • C. Longépé

Métaphore urbaniste — zones, quartiers, îlots, blocs. Particulièrement adaptée à la conception d'un SI cohérent et à un public francophone.

Couches

FR

MÉTHODE PUBLIQUE

Praxeme

Origine française • Praxeme Institute

Méthode holistique et libre de droits, couvrant du sens métier au déploiement. Structurée mais moins répandue que TOGAF.

Holistique

libre

ONTOLOGIE

Zachman

1987 • J. Zachman

Matrice croisant six interrogations (quoi, comment, où, qui, quand, pourquoi) et six perspectives. Une grille de complétude, pas une méthode opérationnelle.

Taxonomie

Le cœur de TOGAF est l'ADM (Architecture Development Method), un cycle de phases qui irrigue toute la suite de ce document. La démarche présentée en Partie 4 en est une transposition simplifiée, calibrée pour des organisations de taille modeste.

2.2 Les normes mobilisables

- ISO/IEC/IEEE 42010:2022** — définit ce qu'est une description d'architecture : parties prenantes, préoccupations, points de vue, vues. Le squelette de tout livrable d'architecture.
- ISO/IEC 25010:2023** — modèle de qualité logicielle. Neuf caractéristiques produit (dont la sécurité et, depuis 2023, la sûreté). Fournit la liste des exigences non fonctionnelles à cadrer.
- ISO/IEC 27001:2022** — management de la sécurité de l'information. Amendée en 2024 pour intégrer la prise en compte du changement climatique dans le contexte de l'organisation.
- BPMN 2.0.2** — notation de modélisation des processus métier, également publiée comme norme ISO/IEC 19510.

La norme ISO/IEC 25010 mérite une attention particulière : ses neuf caractéristiques — adéquation fonctionnelle, performance, compatibilité, capacité d'interaction, fiabilité, sécurité, maintenabilité, flexibilité et sûreté — constituent une liste de contrôle directement utilisable pour cadrer les exigences non fonctionnelles d'un projet et, plus loin, pour évaluer des solutions.

2.3 Quel cadre pour quelle échelle

QUEL NIVEAU DE FORMALISME SELON LA TAILLE DE L'ORGANISATION		
CONTEXTE	FORMALISME RECOMMANDÉ	CADRES MOBILISÉS
TPE / PME	Léger — l'essentiel cadré, pas de cérémonie	ADM picoré, urbanisation simplifiée, ADR
ETI	Intermédiaire — cible et trajectoire formalisées	TOGAF partiel, ArchiMate, ISO 25010
Grand compte	Complet — gouvernance d'architecture dédiée	TOGAF intégral, référentiel outillé

▲ **Sur-ingénierie**

Appliquer TOGAF dans son intégralité à une PME est une erreur classique : le cadre est conçu pour des organisations dotées d'équipes d'architecture dédiées. La valeur, à petite échelle, réside dans le vocabulaire et la séquence de l'ADM — pas dans sa cérémonie complète.

◆ **À retenir**

Les cadres se picorent. La question n'est pas « quel cadre adopter », mais « quel niveau de formalisme le contexte justifie ».

Partie 3 — Modéliser : langages et diagrammes de flux

Modéliser, c'est produire les vues évoquées en Partie 1. Encore faut-il employer la bonne notation pour le bon objet — un point particulièrement source de confusion autour de l'expression « diagramme de flux ».

3.1 « Diagramme de flux » : un terme, plusieurs réalités

« Diagramme de flux » ne désigne pas une notation unique mais une famille d'objets distincts, qui n'appartiennent pas à la même couche et ne s'expriment pas avec la même notation.

« DIAGRAMME DE FLUX » : DE QUOI PARLE-T-ON EXACTEMENT			
CE QU'ON VEUT REPRÉSENTER	TYPE DE DIAGRAMME	NOTATION	COUCHE
L'enchaînement d'un processus métier	Diagramme de processus	BPMN 2.0.2	Métier
Les échanges entre applications	Diagramme de flux applicatif	ArchiMate, schéma d'échange	Applicative
La circulation de la donnée	Flux de données	DFD, traçabilité (lineage)	Données
Une séquence d'interactions techniques	Diagramme de séquence	UML	Applicative / technique

Une précision s'impose sur le DFD (Data Flow Diagram). Cette notation, issue de l'analyse structurée des années 1970–1980, reste pédagogiquement utile pour expliquer la circulation de la donnée, mais elle n'est plus un standard vivant. En 2026, les flux de données se documentent plutôt via les diagrammes d'échange applicatif, le langage ArchiMate ou les outils de traçabilité de la donnée.

3.2 Les langages de modélisation

PROCESSUS

BPMN 2.0.2

2013 • OMG / ISO 19510

Standard de modélisation des processus métier, lisible aussi bien par les métiers que par les techniques.

Métier flux

LOGICIEL

C4 model

Simon Brown

Documentation d'architecture logicielle à quatre niveaux de zoom : contexte, conteneurs, composants, code.

Léger moderne

ENTREPRISE

ArchiMate 3.2

2022 • The Open Group

Langage de modélisation d'architecture d'entreprise, aligné sur TOGAF, reliant métier, applications et technologie.

EA vues

DONNÉES

Merise

France • MCD / MLD / MPD

Méthode de modélisation des données par niveaux conceptuel, logique et physique. Toujours pertinente pour le schéma de données.

Données FR

LOGICIEL

UML

OMG

Famille de diagrammes pour la conception logicielle — classes, séquences, cas d'usage. Complémentaire de BPMN.

Conception

DÉCISION

ADR

2011 • M. Nygard

Architecture Decision Record. Fiche courte traçant une décision : contexte, choix retenu, alternatives, conséquences.

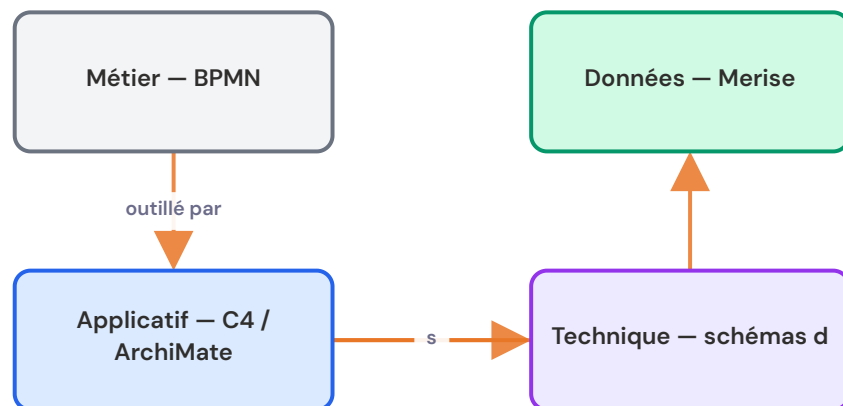
Traçabilité

3.3 Les outils

Ces langages s'appuient sur des outils éprouvés et largement gratuits : Archi pour ArchiMate, Structurizr ou PlantUML pour le modèle C4, ou encore des éditeurs génériques comme draw.io pour les schémas libres. L'outil importe moins que la rigueur de la notation : un diagramme cohérent vaut mieux qu'un bel outil mal employé.

3.4 Où se place chaque modèle

PLACEMENT DES MODÈLES DANS LES DOMAINES D'ARCHITECTURE



◆ À retenir

Avant de tracer un « diagramme de flux », il faut savoir lequel : processus métier, échange applicatif ou circulation de la donnée. Chacun a sa notation et sa couche.

Partie 4 — La démarche : par quoi on commence, par quoi on finit

C'est le cœur du sujet. Une architecture se conçoit selon une séquence ordonnée, encadrée par deux activités transverses, et débouchant sur un plan actionnable.

4.1 Le principe fondateur : le métier avant la technologie

THÈSE

Une architecture SI se conçoit à partir du besoin métier, jamais à partir d'une technologie choisie d'avance.

PARCE QUE

Une technologie est un moyen ; partir du moyen revient à résoudre un problème que l'on n'a pas formulé.

Le besoin métier survit aux technologies, qui se périment ; l'architecture doit donc se raccrocher à ce qui dure.

Un choix posé avant le cadrage devient un invariant non discutable qui contamine toutes les décisions suivantes.

Aligner l'architecture sur les capacités métier permet d'arbitrer coûts et priorités sur des critères objectifs.

MAIS

Une contrainte préexistante — parc installé, obligation réglementaire, compétences internes — est légitime, mais elle doit être posée comme une contrainte explicite du cadrage, pas comme la finalité du projet.

4.2 Deux activités transverses à tout le cycle

Avant même la première phase, deux activités encadrent l'ensemble du cycle et ne s'arrêtent jamais.



Les principes d'architecture

Règles directrices posées en amont — réutiliser avant d'acheter avant de développer, interopérabilité par standards ouverts, souveraineté de la donnée. Elles cadrent toutes les décisions ultérieures.



La gestion des exigences

Activité continue, au centre du cycle et non une phase isolée : recueillir, qualifier et tracer les exigences métier et non fonctionnelles, qui irriguent chaque phase.

4.3 Le cycle en phases

P0

Cadrage

Besoins métier, capacités, parties prenantes, principes et exigences. C'est ici que toute solution posée d'avance est remise en question.

P1

État des lieux

Cartographie de l'existant — processus, applications, données, infrastructure. La situation existante, dite as-is.

P2

Architecture cible

Conception des domaines métier, données, applicatif et technique, sécurité comprise. La situation cible, dite to-be.

P3

Choix techniques

Sélection outillée et justifiée des technologies et logiciels répondant à la cible.

P4

Trajectoire

Plan de transformation par paliers, déduit de l'analyse d'écart entre l'existant et la cible.

P5**Spécification**

Cahier des charges fonctionnel puis technique, base contractuelle de la mise en œuvre.

4.4 L'analyse d'écart, moteur de la trajectoire

Entre l'état des lieux et la cible, un mécanisme produit la trajectoire : l'**analyse d'écart** (gap analysis). Elle consiste à comparer, domaine par domaine, ce qui existe et ce qui est visé, puis à transformer chaque écart en action. C'est cette analyse, et non la cible elle-même, qui rend le projet exécutable.

◆ **Le pont entre la cible et l'action**

La cible décrit où l'on veut aller. L'existant décrit d'où l'on part. L'analyse d'écart décrit comment franchir la distance — c'est elle qui se traduit en trajectoire.

4.5 Détail par phase

ENTRÉES ET LIVRABLES PAR PHASE

PHASE	ENTRÉES	LIVRABLES
P0 — Cadrage	Stratégie, parties prenantes, contraintes	Note de cadrage, principes, exigences
P1 — État des lieux	SI existant, documentation, entretiens	Cartographie de l'existant
P2 — Architecture cible	Cadrage et état des lieux	Architecture cible (4 domaines + sécurité)
P3 — Choix techniques	Cible, exigences non fonctionnelles	Matrice de décision, ADR
P4 — Trajectoire	Écart entre cible et existant	Trajectoire de transformation par paliers
P5 — Spécification	Cible et trajectoire validées	CDC fonctionnel et technique

À chaque phase, la sécurité est traitée dès la conception et non rajoutée ensuite : exigences de sécurité dès le cadrage, intégration aux quatre domaines en phase cible, critères de sécurité dans la sélection technique.

4.6 Le début et la fin, explicitement

CE PAR QUOI ON COMMENCE, CE PAR QUOI ON FINIT

- › On commence toujours par le besoin métier et par l'état de l'existant — jamais par la technologie.
- › On conçoit la cible domaine par domaine, en intégrant la sécurité dès la conception.
- › On mesure l'écart entre la cible et l'existant pour en déduire la trajectoire.
- › On finit par un plan de mise en œuvre par paliers et des spécifications — pas par un simple schéma.

◆ À retenir

On commence par le besoin et l'existant ; on finit par une trajectoire actionnable et des spécifications. Le schéma cible n'est qu'une étape intermédiaire, jamais le livrable final.

Partie 5 — Itérer et gouverner

Une architecture n'est pas un document que l'on produit une fois pour le classer. C'est un actif vivant qui se gouverne et se réajuste.

5.1 Une cible et une trajectoire, pas un livrable figé

Concevoir une architecture, ce n'est pas figer un état définitif, mais tenir simultanément une cible — vers où l'on va — et une trajectoire — comment on y avance, par incréments. La cible se réajuste à mesure que le métier évolue ; la trajectoire se réordonne selon les priorités et les moyens. Une architecture qui prétend tout figer d'un coup est une architecture qui sera obsolète avant d'être mise en œuvre.

5.2 Le raffinement par paliers et la gouvernance d'architecture

Au sein d'un projet, on procède du macro vers le détail : une vision d'ensemble d'abord, puis une descente progressive, couche par couche, validée à chaque jalon. Cette validation est le rôle de la gouvernance d'architecture — une instance, même légère dans une petite structure, qui arbitre les écarts par rapport aux principes et autorise le passage d'un palier au suivant.

5.3 Le rôle de l'architecte et le référentiel

L'architecture est portée par un rôle — l'architecte — qui fait l'interface entre les parties prenantes métier et les équipes techniques. Son travail ne vaut que s'il est consigné dans un référentiel maintenu : la cartographie du SI et les décisions prises doivent rester accessibles et à jour. Un référentiel laissé à l'abandon vaut moins qu'aucun référentiel, car il induit en erreur.

5.4 L'architecture évolutive

Les approches modernes proposent de rendre l'évolutivité mesurable, au moyen de fonctions d'aptitude (fitness functions) — des contrôles automatisés qui vérifient en continu que l'architecture respecte ses propriétés clés (performance, couplage, sécurité). Couplées à des décisions tracées et

régulièrement revues, elles permettent à l'architecture de se réajuster sans dériver.

5.5 Les trois échelles d'itération

TROIS ÉCHELLES D'ITÉRATION

- › Intra-projet : raffinement du macro vers le détail, validé à chaque jalon.
- › Inter-version : à chaque évolution majeure du métier ou de la technologie, on repasse dans le cycle.
- › Cycle de vie : la cartographie et les décisions sont maintenues vivantes, pas archivées une fois pour toutes.

◆ À retenir

L'architecture s'itère à trois échelles. Ce qui ne se réajuste pas se périmé — y compris la cartographie et les décisions.

Partie 6 — Choisir les outils et les logiciels

Les choix techniques arrivent tard dans la démarche, et ce n'est pas un hasard : ils découlent de la cible et des exigences, jamais l'inverse. Encore faut-il les conduire avec méthode.

6.1 La méthode de décision rationnelle

SÉLECTIONNER UNE TECHNOLOGIE DE FAÇON RATIONNELLE

- › Exprimer le besoin et le contexte avant de regarder les solutions.
 - › Définir des critères pondérés, en s'appuyant sur les caractéristiques d'ISO/IEC 25010.
 - › Évaluer chaque candidat sur ces critères, sans complaisance.
 - › Comparer en intégrant le coût total de possession (TCO), pas seulement le prix d'entrée.
 - › Tracer la décision dans un ADR, avec les alternatives écartées et leurs raisons.
-

6.2 Évaluer un logiciel open source : QSOS

QSOS — Qualification et Sélection de logiciels Open Source : méthode d'évaluation libre (licence GFDL) conçue par Atos. Structurée en quatre étapes itératives, elle évalue les logiciels selon trois axes — couverture fonctionnelle, risques du point de vue de l'utilisateur, risques du point de vue du prestataire de services.

Un logiciel libre reste un logiciel : il s'évalue. Mais il appelle des questions spécifiques que QSOS formalise — la pérennité de la structure porteuse, l'activité de la communauté, le type de licence, la qualité des développements. La méthode procède en définissant un référentiel de critères, en évaluant les candidats sur ces critères, en qualifiant le besoin par pondération, puis en sélectionnant. Recourir à une grille reconnue plutôt qu'à une impression personnelle est ce qui distingue un choix défendable d'un pari.

6.3 Tracer la décision : l'ADR

ADR-012

SGBD relationnel open source plutôt qu'une base propriétaire

ACTÉE

2026-06-15 · Architecte et direction

JUSTIFICATION

Besoin transactionnel fort, souveraineté de la donnée, absence de coût de licence, écosystème mature et compétences internes existantes.

ALTERNATIVE ÉCARTÉE

Base propriétaire écartée — coût de licence récurrent et dépendance éditeur non justifiés par les besoins exprimés.



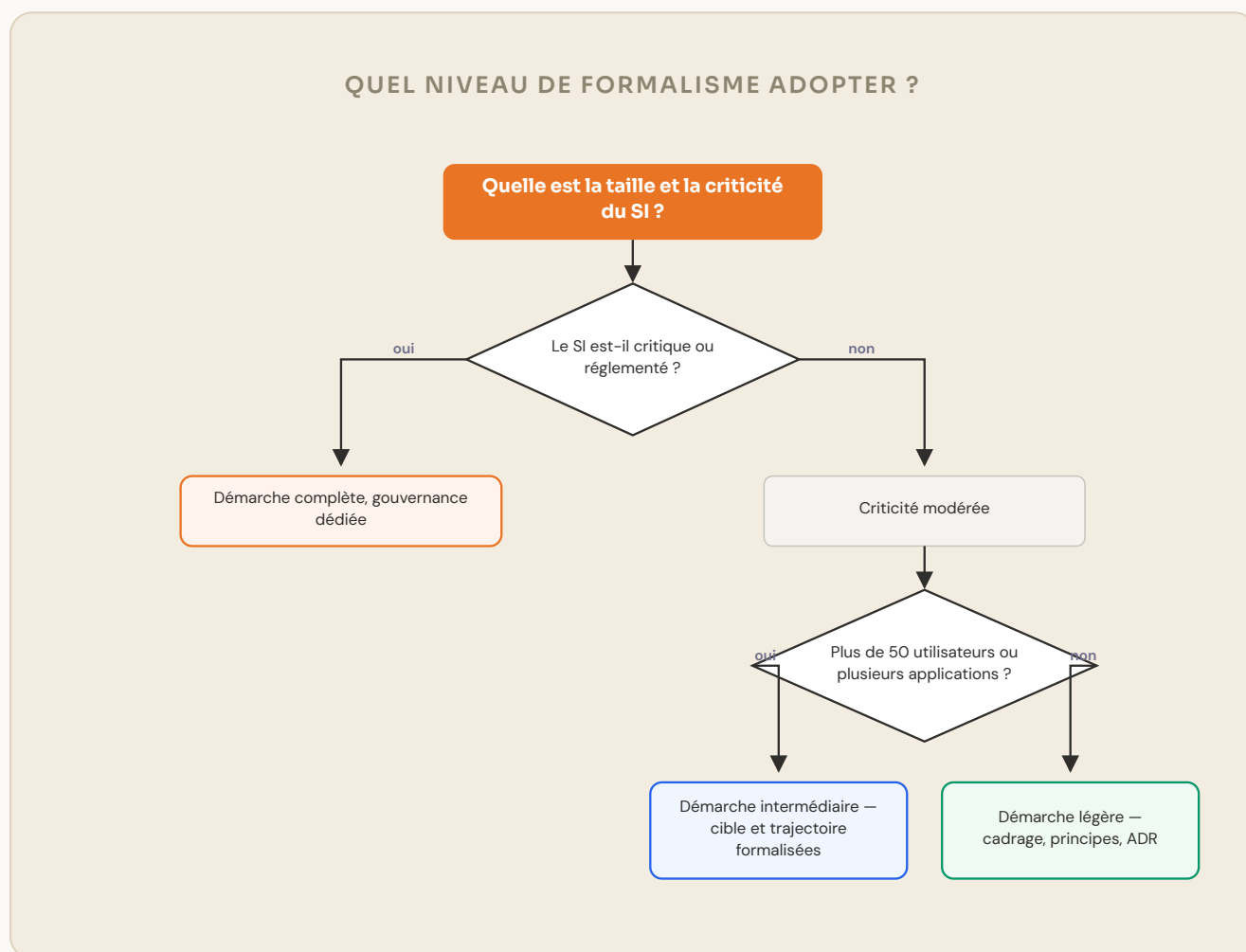
À retenir

On ne choisit pas un outil au feeling. On exprime le besoin, on pondère des critères, on intègre le coût total, et on trace la décision — afin qu'elle reste compréhensible et révisable plus tard.

Guide décisionnel

Cette section condense la démarche en trois aides à la décision : un arbre pour choisir le niveau de formalisme, une matrice pour choisir un cadre, et des scénarios types.

Arbre de décision



Matrice cadre × contexte

QUEL CADRE SELON LE CONTEXTE			
CONTEXTE	URBANISATION	TOGAF	ARCHIMATE

Conception d'un SI à partir de zéro	Adapté	Partiel	Utile
Transformation d'un grand SI existant	Utile	Adapté	Adapté
PME, besoin pragmatique	Simplifié	À picorer	Optionnel
Modélisation liée métier-technologie	Limité	Partiel	Adapté

Scénarios types

1 Startup / TPE

- ✓ Contrainte : aller vite, ressources limitées
- ✓ Démarche légère : cadrage, principes, ADR
- ✓ Pas de référentiel lourd

2 PME

- ✓ Contrainte : éviter la sur-ingénierie
- ✓ Cible et trajectoire formalisées, sans cérémonie
- ✓ Urbanisation simplifiée

3 ETI

- ✓ Contrainte : maîtriser un SI déjà complexe
- ✓ TOGAF partiel et ArchiMate
- ✓ Gouvernance d'architecture émergente

4 Secteur public / réglementé

- ✓ Contrainte : conformité et traçabilité
- ✓ Démarche complète, sécurité et souveraineté dès la conception
- ✓ Référentiel d'architecture maintenu

Tendances et horizon 2026

La discipline évolue. Les tendances ci-dessous, observées chez les responsables d'architecture en 2026, méritent d'être surveillées — non pour les suivre aveuglément, mais pour anticiper leur impact.



L'IA se gouverne, elle ne se centralise pas

Priorité numéro un des responsables d'architecture en 2026. L'IA, y compris agentique, devient un composant du SI à gouverner — pas un socle autour duquel tout se construit.



Souveraineté numérique dès la conception

La maîtrise de la localisation et des accès à la donnée se conçoit dans l'architecture, pas après coup.



Sécurité embarquée

La sécurité s'affirme comme une préoccupation transverse, intégrée à chaque couche et de plus en plus intriquée avec l'adoption de l'IA.



Platform engineering

Les plateformes internes de développement — parcours balisés, libre-service — deviennent un standard d'organisation, au-delà du seul DevOps.



Data-as-a-Product

La donnée traitée comme un produit — propriétaire, contrat de service, schéma, interface — plutôt que comme un sous-produit dispersé.



Sobriété numérique

La consommation énergétique devient une contrainte d'ingénierie de premier ordre, sous l'effet de la charge induite par l'IA et de la pression sur les coûts.

i Recommandation prospective

Pour toute organisation tentée de bâtir son SI « autour d'une IA » : inverser la logique. Préparer une architecture saine — données gouvernées, sécurité et souveraineté intégrées — dans laquelle l'IA s'insère comme un composant. Une intelligence artificielle posée sur des fondations fragiles reste fragile.

Conclusion

Trois idées traversent ce document. La première : une architecture SI part du besoin métier et de l'existant, jamais d'une technologie choisie d'avance. La deuxième : elle se décline en domaines — métier, données, applicatif, technique — traversés par la sécurité, et se décrit non par un schéma unique mais par un ensemble de vues adressant les préoccupations des parties prenantes. La troisième : elle se conçoit selon une séquence claire, du cadrage à la trajectoire, et se maintient vivante par itérations.

La valeur d'une architecture ne tient pas à l'épaisseur de sa documentation, mais à sa capacité à éclairer des décisions et à rester actionnable. Le cadre se picore, le formalisme s'adapte à l'échelle, et chaque décision se trace pour rester révisable. C'est à cette condition qu'un système d'information devient un actif maîtrisé plutôt qu'une dette qui s'accumule.

CRITIQUE

Cadrer le besoin métier avant tout choix de technologie.

IMPORTANT

Maintenir l'architecture vivante — une cible, une trajectoire, des décisions tracées.

UTILE

Outils la modélisation et le référentiel pour faciliter la gouvernance.

Références et ressources

Cadres et méthodes

- TOGAF Standard, 10th Edition (The Open Group) : www.opengroup.org ↗
- Urbanisation du SI (C. Longépé) : www.dunod.com ↗
- Praxeme — méthode publique : www.praxeme.org ↗
- C4 model (Simon Brown) : c4model.com ↗
- Architecture Decision Records (M. Nygard) : adr.github.io ↗

Normes

- ISO/IEC/IEEE 42010:2022 — Architecture description : www.iso.org ↗
- ISO/IEC 25010:2023 — Product quality model : www.iso.org ↗
- ISO/IEC 27001:2022 — Information security : www.iso.org ↗
- BPMN 2.0.2 / ISO/IEC 19510 (OMG) : www.omg.org ↗
- ArchiMate 3.2 (The Open Group) : pubs.opengroup.org ↗

Méthode open source

- QSOS — Qualification et Sélection de logiciels Open Source : www.qsos.org ↗

Ouvrages

- Fundamentals of Software Architecture — Mark Richards, Neal Ford (O'Reilly)
- Software Architecture in Practice — Bass, Clements, Kazman (SEI)
- The Practice of Enterprise Architecture — Svyatoslav Kotusev
- Le projet d'urbanisation du SI — Christophe Longépé (Dunod)
- Designing Data-Intensive Applications — Martin Kleppmann (O'Reilly)
- Building Evolutionary Architectures — Ford, Parsons, Kua (O'Reilly)
- Clean Architecture — Robert C. Martin (Prentice Hall)

Glossaire

Architecture d'entreprise

Discipline englobant la stratégie, le métier et le système d'information d'une organisation. L'architecture SI en est un sous-ensemble.

Architecture SI

Organisation d'ensemble reliant les processus métier aux applications, aux données et à l'infrastructure.

Urbanisation

Approche, d'origine française, qui structure le SI par analogie avec l'aménagement d'une ville (zones, quartiers, îlots, blocs).

Capacité métier

Aptitude d'une organisation à réaliser une activité, indépendamment de la manière dont elle est outillée.

Point de vue / vue

Manière convenue de représenter une architecture pour adresser une préoccupation (point de vue) ; le résultat concret de son application (vue).

Principe d'architecture

Règle directrice posée en amont, qui cadre l'ensemble des décisions ultérieures.

ADM

Architecture Development Method — le cycle de conception en phases au cœur de TOGAF.

ADR

Architecture Decision Record — fiche traçant une décision d'architecture, ses alternatives et ses conséquences.

Analyse d'écart

Comparaison entre l'existant et la cible, dont découle la trajectoire de transformation.

BPMN

Business Process Model and Notation — notation standard de modélisation des processus métier.

C4

Modèle de documentation d'architecture logicielle à quatre niveaux : contexte, conteneurs, composants, code.

ArchiMate

Langage de modélisation d'architecture d'entreprise, aligné sur TOGAF.

Merise

Méthode française de modélisation des données par niveaux conceptuel, logique et physique.

TCO

Total Cost of Ownership — coût total de possession d'une solution sur son cycle de vie, au-delà du prix d'acquisition.

Fonction d'aptitude

Contrôle automatisé (fitness function) vérifiant en continu qu'une architecture respecte une propriété clé.

Historique des révisions

VERSION	DATE	AUTEUR	MODIFICATIONS
v1.0.0	21 juin 2026	Mattieu Pottier	Version initiale

À propos de l'auteur

Mattieu Pottier

Consultant indépendant en transformation digitale — MP-i · Mattieu Pottier Indépendant

Consultant freelance spécialisé en ERP Odoo, SIG (QGIS, PostGIS), applications web et automatisation. Intervient sur des projets de cadrage, de conception d'architecture, de développement et d'accompagnement au changement pour des PME, ETI et collectivités.

mp-i.pro

MP-i — Mattieu Pottier Indépendant