# Urbanisation des SI

(Enterprise Architecture 🇬🇧)

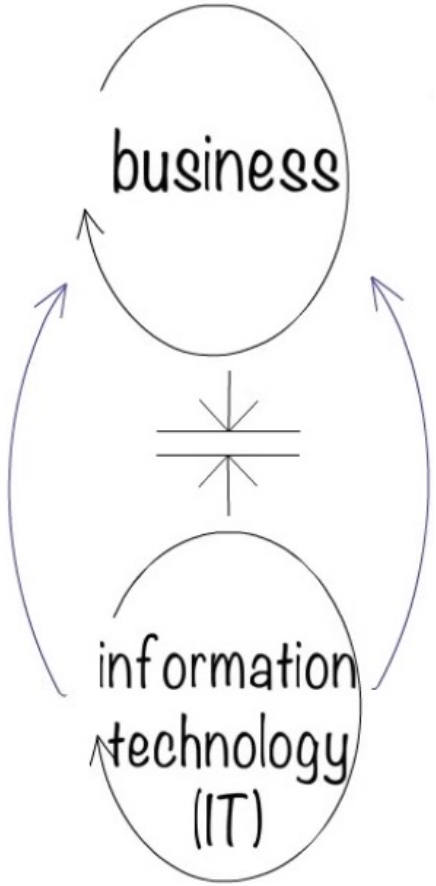
# Sommaire

1. Objectifs de l’EA
2. Pourquoi l’EA
3. Comment la mettre en place
    1. Roadmap de transition
    2. Préparer la transition via modèle CSVLOD
    3. CSVLOD Artifacts IT
        - Standards
        - Landscape
        - Design

# 1. Objectifs

# 🎯 Objectif de l’Enterprise Architecture

- Aligner stratégie et systèmes d’information
- Structurer l’évolution du SI
- Optimiser les processus et ressources

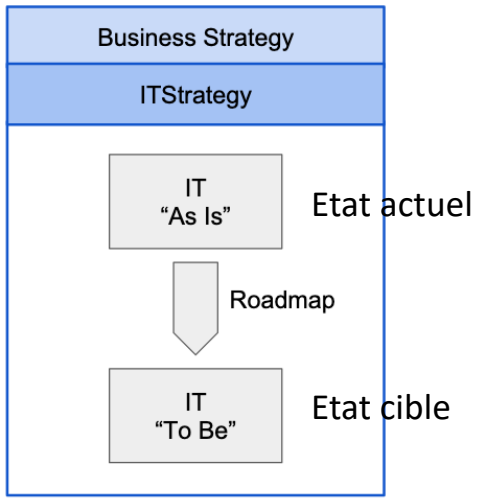


# 2. Pourquoi

# EA Pourquoi ?

**Le monde évolue rapidement l'entreprise doit évoluer <u>rapidement</u>**

- Faciliter la prise de décision
- Faire des investissements technologiques qui soutiennent le business
- Permettre la transformation (e.g. transition roadmap)



# 3. Comment

# EA Comment ?

- Un plan qui lie 4 couches
  - Business
  - Application
  - Data
  - Technologie

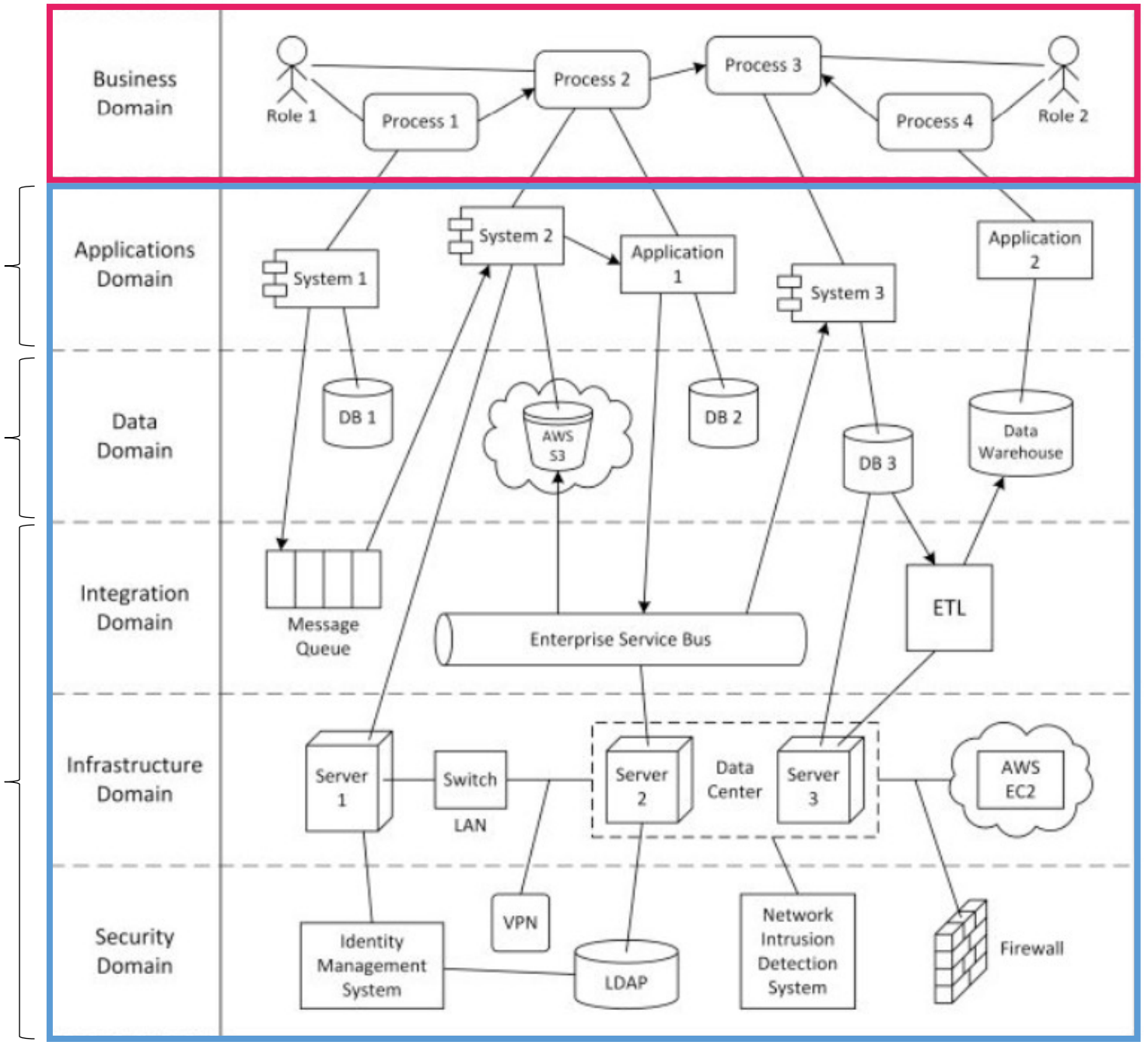
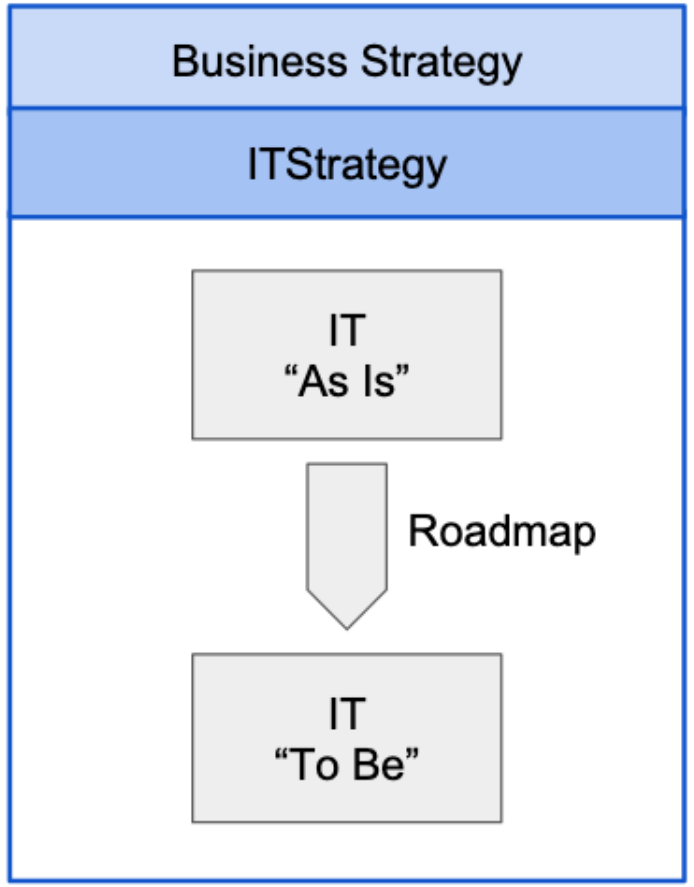



**Figure 2.2. EA domains as different layers of an organization**

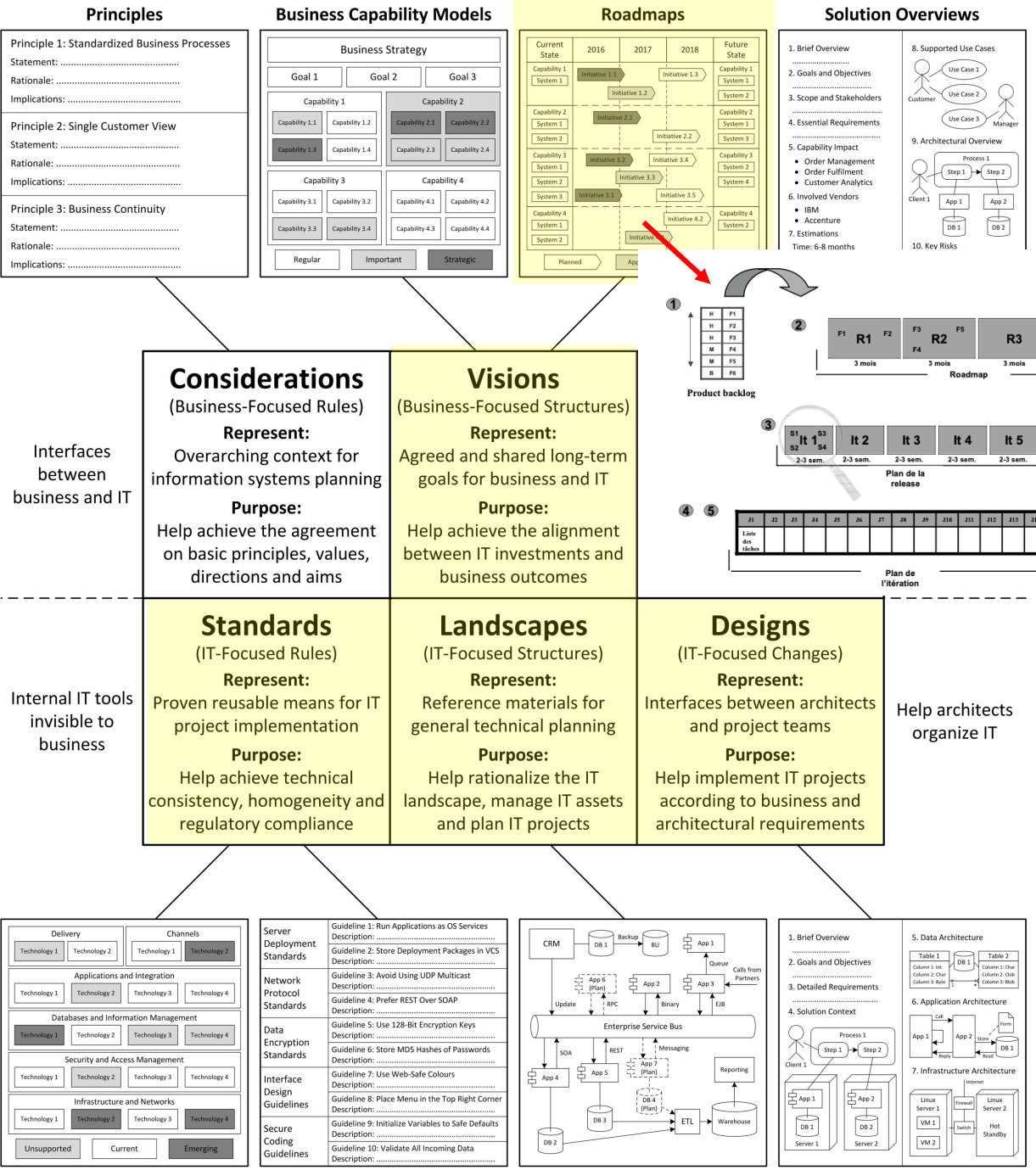
# 3.1 Roadmap de transition

# Passer de l’état actuel à l’état cible

1. **Diagnostic & cartographie de l’existant**
   - Fonctionnement de l’entreprise : cartographie du SI
   - Identifier douleurs : silos, coûts, lenteurs …
2. **Définition de l’état cible**
   - Quel paysage applicatif (SaaS, orienté cloud)
   - Quelle gestion des données (RGPD, stockage)
   - Quelles techno (API, event, sécurité)
3. **Feuille de route**
   - Découpage en phases : court, moyen, long terme
   - Estimation des budgets, ressources, risques, KPIs



# 3.2 Préparer roadmap de transition modèle CSVLOD




# Modèle CSVLOD

- les Considerations et Visions expriment ce qui est important pour le business,
- les Standards et Landscapes structurent la réponse IT,
- les Outlines et Designs traduisent cette vision en projets concrets.

Svyatoslav Kotusev (kotusev@kotusev.com) for the British Computer Society (BCS)

# 3.2 CSVLOD Artifacts IT

| | Standards | Landscapes | Designs |
|---|---|---|---|
| IT-Focused | Help achieve technical consistency, technological homogeneity and regulatory compliance | Help understand, analyze and modify the structure of the IT landscape | Help implement approved IT projects according to business and architectural requirements |

# Standards

*Comment voulons-nous faire les choses de manière homogène dans l'entreprise ?*

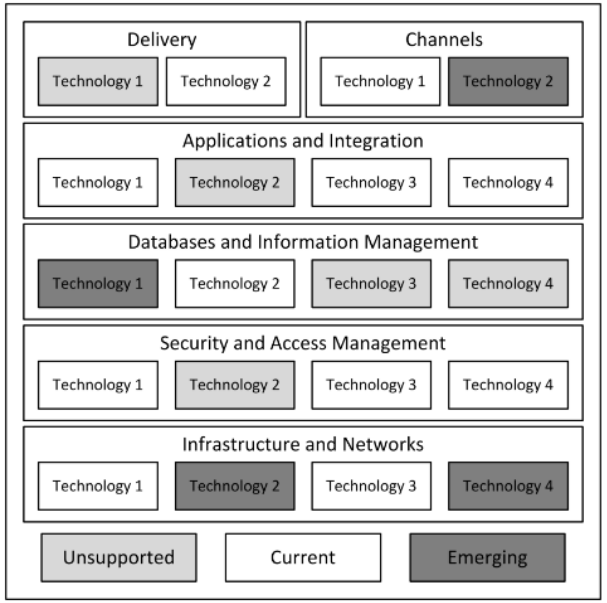
**Objectifs :**

- Conformité règlementaire (RGPD, ISO)
- Homogénéiser le SI

# 4 principaux niveaux de standards

- Standards **techniques** : (SaaS)
- Standards **d'intégration** : (api)
- Standards **data** :
- Standards **sécurité / compliance** (oauth, RGPD)

## Technology Reference Models




## Guidelines

| | |
|---|---|
| Server Deployment Standards | Guideline 1: Run Applications as OS Services<br>Description: ..... |
| | Guideline 2: Store Deployment Packages in VCS<br>Description: ..... |
| Network Protocol Standards | Guideline 3: Avoid Using UDP Multicast<br>Description: ..... |
| | Guideline 4: Prefer REST Over SOAP<br>Description: ..... |
| Data Encryption Standards | Guideline 5: Use 256-Bit Encryption Keys<br>Description: ..... |
| | Guideline 6: Store MD5 Hashes of Passwords<br>Description: ..... |
| Interface Design Guidelines | Guideline 7: Use Web-Safe Colours<br>Description: ..... |
| | Guideline 8: Place Menu in the Top Right Corner<br>Description: ..... |
| Secure Coding Guidelines | Guideline 9: Initialize Variables to Safe Defaults<br>Description: ..... |
| | Guideline 10: Validate All Incoming Data<br>Description: ..... |

# Exemples

| Standards | Objectifs | Référence ISO indicative |
|---|---|---|
| Le code source doit être versionné dans un SCM centralisé | Traçabilité des changements, intégrité, auditabilité | ISO/IEC 27002:2022<br>- 8.32 (gestion du changement),<br>- 8.4 (accès au code source) ; |

## 8.4 Accès aux codes source

| Type de mesure de sécurité | Propriétés de sécurité de l'information | Concepts de cybersécurité | Capacités opérationnelles | Domaines de sécurité |
|---|---|---|---|---|
| #Préventive | #Confidentialité<br>#Intégrité<br>#Disponibilité | #Protéger | #Gestion_des_identités_et_des_accès<br>#Sécurité_des_applications<br>#Configuration_sécurisée | #Protection |

**Mesure de sécurité**

Il convient de gérer de manière appropriée l'accès en lecture et en écriture au code source, aux outils de développement et aux bibliothèques de logiciels.

**Objectif**

Empêcher l'introduction d'une fonctionnalité non autorisée, éviter les modifications non intentionnelles ou malveillantes et préserver la confidentialité de la propriété intellectuelle importante.

**Recommandations**

Il convient de contrôler de manière stricte l'accès aux codes source et aux éléments associés (tels que conceptions, spécifications, plans de vérification et de validation) et aux outils de développement (par exemple, compilateurs, générateurs, outils d'intégration, plateformes de test et environnements).

En ce qui concerne les codes source, ceci peut être réalisé en contrôlant le stockage central d'un code, de préférence dans le système de gestion du code source.

L'accès en lecture et l'accès en écriture aux codes source peuvent différer selon la fonction du personnel. Par exemple, l'accès en lecture aux codes source peut être largement fourni au sein de l'organisation, mais l'accès en écriture aux codes source est seulement accordé à des employés privilégiés ou à des propriétaires désignés. Lorsque des composants d'un code sont utilisés par plusieurs développeurs au sein d'une organisation, il convient de mettre en œuvre un accès en lecture à un répertoire de code centralisé. De plus, si des composants d'un code source libre ou d'un code de tierces parties sont utilisés dans une organisation, l'accès en lecture à ces répertoires de code externe peut être largement fourni. Cependant, il convient que l'accès en écriture soit toujours restreint.

Il convient de prendre en considération les lignes directrices suivantes pour contrôler l'accès aux bibliothèques de codes source des programmes afin de réduire la possibilité d'altération des programmes informatiques:

a) gérer l'accès aux codes source des programmes et aux bibliothèques des codes source de programmes conformément aux procédures établies;

b) attribuer l'accès en lecture et en écriture aux codes source en fonction des besoins métier et le gérer pour traiter les risques d'altération ou d'utilisation abusive et conformément aux procédures établies;

c) mettre à jour le code source et les éléments associés et attribuer l'accès au code source conformément aux procédures de contrôle des changements (voir 8.32) et réaliser l'attribution d'accès seulement après avoir reçu l'autorisation appropriée;

d) ne pas accorder aux développeurs un accès direct au répertoire de code source, mais à travers des outils de développement qui contrôlent les activités et les autorisations sur le code source;

e) garder les listings des programmes dans un environnement sécurisé, où il convient que les accès en lecture et en écriture soient gérés et attribués de manière appropriée;

f) tenir un journal d'audit de tous les accès et de toutes les modifications apportées au code source.

Si le code source du programme est destiné à être publié, il convient d'envisager des mesures de sécurité supplémentaires pour apporter l'assurance de son intégrité (par exemple, signature électronique).

## 8.32 Gestion des changements

| Type de mesure de sécurité | Propriétés de sécurité de l'information | Concepts de cybersécurité | Capacités opérationnelles | Domaines de sécurité |
|---|---|---|---|---|
| #Préventive | #Confidentialité<br>#Intégrité<br>#Disponibilité | #Protéger | #Sécurité_des_applications<br>#Sécurité_système_et_réseau | #Protection |

**Mesure de sécurité**

Il convient que les changements apportés aux moyens de traitement de l'information et aux systèmes d'information soient soumis à des procédures de gestion des changements.

**Objectif**

Préserver la sécurité de l'information lors de l'exécution des changements.

**Recommandations**

Il convient que l'introduction de nouveaux systèmes et de changements importants apportés aux systèmes existants suivent les règles convenues et un processus formel de documentation, de spécification, de tests, de contrôle qualité et de mise en œuvre gérée. Il convient que des responsabilités et des procédures de management soient mises en place pour assurer un contrôle satisfaisant de tous les changements.

Il convient que les procédures de contrôle des changements soient documentées et appliquées afin d'assurer la confidentialité, l'intégrité et la disponibilité des informations dans les moyens de traitement de l'information et les systèmes d'information, pendant le cycle de vie de développement de tout le système, depuis les étapes de conception initiales jusqu'aux activités de maintenance ultérieures.

Si possible, il convient que les procédures de contrôle des changements pour l'infrastructure TIC et les logiciels soient intégrées.

Il convient que les procédures de contrôle des changements incluent:

a) la planification et l'évaluation des impacts potentiels des changements en prenant en compte toutes les dépendances;

b) l'autorisation des changements;

c) la communication des changements aux parties intéressées pertinentes;

d) les tests et l'acceptation des tests relatifs aux changements (voir 8.29);

e) la mise en œuvre des changements, y compris les plans de déploiement;

f) les considérations d'urgence et de secours, y compris les procédures de repli;

g) le maintien à jour des enregistrements des changements, qui incluent tout ce qui précède;

h) l'assurance que la documentation de fonctionnement (voir 5.37) et les procédures utilisateurs sont modifiées autant que nécessaire pour rester appropriées;

i) l'assurance que les plans de continuité des TIC et les procédures de réponse et de reprise (voir 5.30) sont modifiés autant que nécessaire pour rester appropriés.

# Exemples

**Objectif**

Protéger l'environnement opérationnel et les données correspondantes contre les compromissions qui pourraient être dues aux activités de développement et de test.

**Recommandations**

Il convient de déterminer et de mettre en œuvre le niveau de séparation nécessaire entre les environnements de développement, de test et opérationnels afin d'empêcher les problèmes opérationnels.

Il convient de prendre en considération les éléments suivants:

a) séparer de façon adéquate les systèmes de développement et de production et les faire fonctionner dans des domaines différents (par exemple, dans des environnements physiques ou virtuels distincts);

b) définir, documenter et mettre en œuvre les règles et autorisations pour le déploiement de logiciels depuis le développement jusqu'à l'état de production;

c) tester les modifications apportées aux systèmes et aux applications opérationnels dans un environnement de test ou de simulation avant de les appliquer aux systèmes opérationnels (voir 8.29);

d) ne pas réaliser de tests dans des environnements opérationnels sauf dans des circonstances définies et approuvées;

e) rendre inaccessibles les compilateurs, éditeurs et autres outils de développement ou programmes utilitaires depuis les systèmes opérationnels lorsqu'ils ne sont pas nécessaires;

f) afficher les marques d'identification de l'environnement appropriées dans les menus afin de réduire les risques d'erreur;

g) ne pas copier d'informations sensibles dans les environnements des systèmes de développement et de test sauf si des mesures de sécurité équivalentes sont mises en place pour les systèmes de développement et de test.

| Standards | Objectifs | Référence ISO indicative |
|---|---|---|
| Les environnements de dev / test / prod sont séparés | Les environnements de dev / test / prod sont séparés | ISO/IEC 27002:2022<br>- 8.31 (Séparation des environnements de développement, de test et opérationnels) |

# Exemples

| Standards | Objectifs | Référence ISO indicative |
|---|---|---|
| Les décisions d'architecture sont documentées et associées à des vues | Alignement avec les besoins des parties prenantes, gouvernance | ISO/IEC/IEEE 42010<br>- 5.2.7 (Architecture views and architecture viewpoints)<br>- 5.2.12 (Architecture decisions and rationale) |

An architecture decision record (ADR) is a document that captures an important architecture decision made along wi

https://github.com/joelparkerhenderson/architecture-decision-record

# Les Normes ISO

Les normes ISO, émises par l'Organisation Internationale de Normalisation, **sont des lignes directrices et des outils conçus pour assurer que les produits, services et systèmes répondent à des critères de qualité, de sécurité et d'efficacité spécifiques**.

# Principales normes ISO en informatique

- ISO/IEC/IEEE 12207 – Software life cycle processes
- ISO/IEC 27001 – Information Security Management System (ISMS)
- ISO/IEC/IEEE 29119 – Software Testing

# Normes ISO et entreprises

- Les normes ISO sont **volontaires par défaut** : une organisation choisit de s'y conformer.
- Elles deviennent **obligatoires** si :
  - la loi d'un pays les rend obligatoires dans un domaine,
  - un contrat ou un appel d'offres l'exige,
  - l'entreprise décide d'obtenir une **certification** (ex. ISO 27001, ISO 9001) et doit alors démontrer sa conformité.

# Norme ISO devient obligatoire en France

- **Exemple : société d'informatique qui héberge des données de santé (HDS) pour des hôpitaux :**
  - **L1111-8 du Code de la santé publique**.
    - *[…] L'hébergeur de données mentionnées au premier alinéa du I sur support numérique est titulaire d'un certificat de conformité.*
  - **Or cette certification se base sur la norme ISO 27001**
    - *L'organisme procède à un audit en deux étapes pour évaluer la conformité de l'hébergeur aux exigences du référentiel de certification. Il vérifie notamment l'équivalence des éventuelles certifications ISO 27001 ou ISO 20000 déjà obtenues par l'hébergeur. (source)*

➡ L'entreprise doit donc se soumettre à un audit

# Landscape

*À quoi ressemble réellement notre système aujourd'hui (et demain) ?*

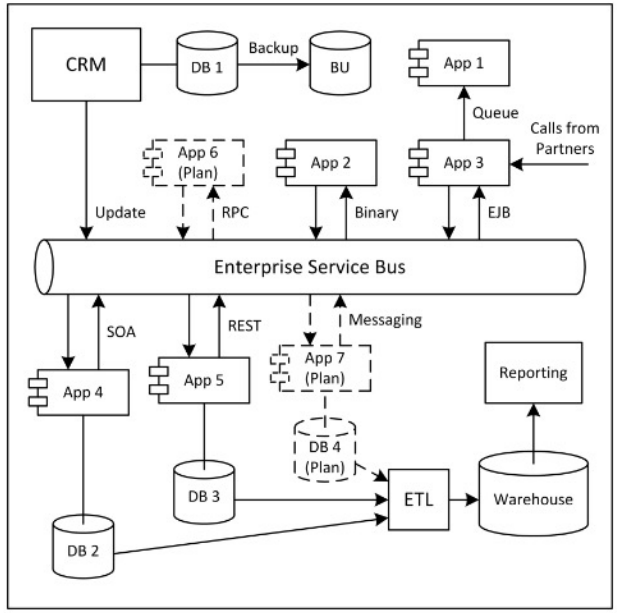
C'est la **cartographie des systèmes** (applications, plateformes, interfaces…).

# Landscape

- **Applications à retirer**
  - Ex : “CRM Legacy” → retrait prévu en 3 vagues
- **Applications à remplacer**
  - Ex : “ERP actuel” → migration vers un ERP Cloud
- **Applications à moderniser**
  - Ex : application monolithique → découpage progressif en services
- **Applications à conserver**
  - Ex : système cœur métier stable, mais à intégrer proprement

## Landscapes are IT-Focused Structures

### Landscape Diagrams




### Inventories

| Asset | Purpose | Owners | Cost | Problems |
|---|---|---|---|---|
| Application 1 | ..... | ..... | ..... | ..... |
| Application 2 | ..... | ..... | ..... | ..... |
| Application 3 | ..... | ..... | ..... | ..... |
| Application 4 | ..... | ..... | ..... | ..... |
| System 1 | ..... | ..... | ..... | ..... |
| System 2 | ..... | ..... | ..... | ..... |
| System 3 | ..... | ..... | ..... | ..... |
| System 4 | ..... | ..... | ..... | ..... |
| System 5 | ..... | ..... | ..... | ..... |
| Database 1 | ..... | ..... | ..... | ..... |
| Database 2 | ..... | ..... | ..... | ..... |
| Database 3 | ..... | ..... | ..... | ..... |
| Database 4 | ..... | ..... | ..... | ..... |

Decommission | Reuse | Invest

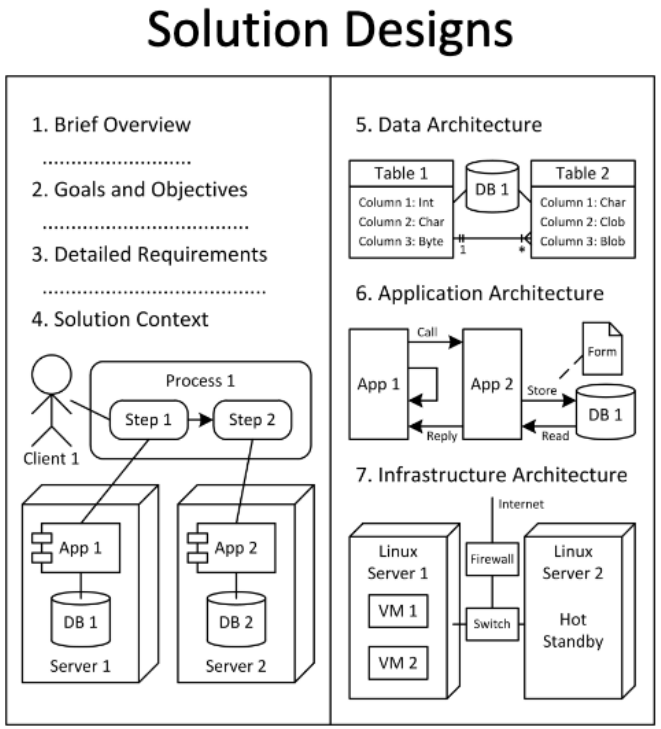
Objectifs : soutenir la prise de décisions techniques et faciliter la planification des projets

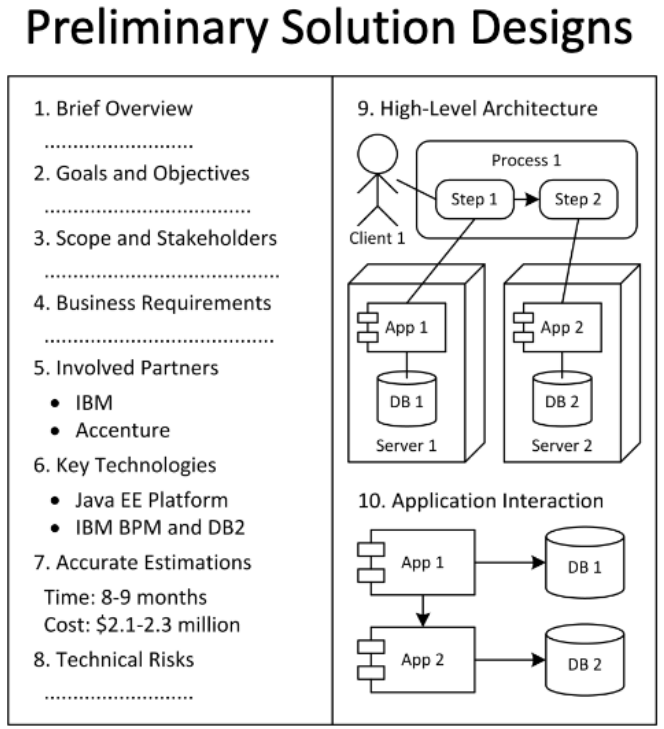
# Design

*Comment concevoir cette solution de manière cohérente avec nos règles (standards) et notre terrain (landscape) et en accord avec la vision produit ?*

# Design

- Le **design**, c'est :
- la **vision cible d'architecture** (macro-design)
- les **patterns structurants** (microservices, événements, CQRS, data mesh, etc.)




Objectifs : mettre en œuvre des projets informatiques conformément aux exigences commerciales et architecturales