1) Référentiel d'architecture cible (vue logique)

Patron recommandé : Lakehouse (zones *Raw/Bronze → Silver → Gold* **+ entrepôt/ lakehouse + services temps réel et ML/AI) avec outillage DataOps/MLOps, gouvernance & sécurité transverses.**

Composants clés & rôle :

1. **Accès & Réseau** : Hub-&-Spoke, Private DNS, Private Endpoints, pare-feu, WAF/ Front Door (zéro exposition publique)
2. **Stockage** : Data Lake (Objets/Delta) + *staging* + *archive* ; versioning & lifecycle policies
3. **Ingestion** : Batch (ELT/ETL), streaming (Kafka/Event Hubs), CDC (Change Data Capture) ; connecteurs SaaS/ERP/CRM
4. **Traitements** : Spark/SQL (batch/stream), jobs planifiés, transformations normalisées (bronze→silver→gold) ; *warehouse* (Synapse SQL/Snowflake) pour BI/serving
5. **Servir/Consommer** : Couches **semantic/BI** (Power BI, DAX/semantic models) et **APIs** via API Management pour l'externe/inter-applicatif
6. **ML/AI** : Registry (modèles), *feature store*, pipelines d'entraînement & déploiement, endpoints sécurisés, *shadow/blue-green* pour les modèles
7. **Orchestration & DataOps** : Orchestrateur (Synapse/ADF/Workflows, Airflow), *jobs* idempotents, reprise sur incident, *CI/CD* des pipelines
8. **Gouvernance & Métadonnées** : **Data Catalog** (glossaire, classification, lineage), politiques d'accès (ABAC/RBAC), *data quality rules*, audits & PIA RGPD
9. **Sécurité & Conformité** : IAM centralisé (Entra ID), chiffrement *at rest* et *in transit*, KMS/Customer-Managed Keys, DLP, masquage/Row-Level/Column-Level security, **logs & SIEM**
10. **Observabilité & FinOps** : métriques SLO/SLA, suivi coûts par *tags* (projet/ domaine), *budgets/alerts*, *data usage telemetry*

2) Patrons de déploiement (Architecture)

2.1 Landing Zone data (fondations)

Séparation des environnements : *DEV* / *TEST* / *PREPROD* / *PROD* avec *RBAC* & quotas dédiés ; naming & *tagging* normalisés (coût, propriétaire, sensibilité).

Topologie réseau : *Hub-&-Spoke* (sécurité centralisée), **Private Link/ Endpoints** vers les PaaS ; interdiction d'IP publiques ; *Just-In-Time* pour bastions.

Identité : Entra ID + groupes synchronisés ; **PIM** pour élévation temporaire ; *managed identities* pour les services.

Clés & Secrets : **Key Vault** (CMK, rotation), *double encryption* si requis (données sensibles).

Journalisation : Routage central vers Log Analytics/SIEM ; rétention conforme (RGPD/archivage).

2.2 Automatisation IaC & CI/CD

IaC (Terraform/Bicep) pour *tous* les artefacts (réseau, stockage, compute, sécurité).

Pipelines CI/CD (Azure DevOps/GitHub) : *plan* → *review* → *apply* avec *policies* de validation et scans de sécurité ; *release* par environnement.

2.3 Zonage Data Lake

- **Bronze (raw)** : ingestion *append-only*, schéma tardif ;
- **Silver (conforme)** : normalisation, qualité, PII traitées ;
- **Gold (serving)** : modèles analytiques/BI, *late binding* des autorisations.

3) Sécurité by-design (contrôles concrets)

3.1 Contrôles préventifs

Least privilege : RBAC/ABAC par *domaines data* ; *deny policies* (pas de PaaS exposé).

Data protection : CMK (HSM), *TLS 1.2*+, *mTLS* pour API internes ; DLP (règles sur exfiltration).

PII & RGPD : classification automatique (catalog), masquage dynamique (RLS/CLS), *data minimization*.

Build security : scans IaC (drift, misconfig), *secret scanning*, SAST/Dependency scanning.

Chiffrement : *at rest* (Storage/Snowflake) + *in transit* ; rotation des clés/secrets (<90 j).

3.2 Contrôles détectifs

- **Audit & Logs** : accès, actions d'admin, *data lineage* ; *risk scoring*.
- **Observabilité** : quotas/latence/temps de run, taux d'échecs ; *anomaly detection* sur coûts/usage.
- **Threat protection** : Defender/CloudTrail-like ; alertes *privileged operations*.

3.3 Contrôles correctifs

- **Break-glass** (comptes d'urgence contrôlés), *runbooks* d'incident data, *auto-remediation* (policies/Functions).
- **Plan de réponse** : RTO/RPO, *backup immutable*, *point-in-time restore* (warehouse).

4) Orchestration & DataOps/MLOps — déploiement

- **Pipelines d'ingestion** : *source* → *raw* (ACID/Delta), *idempotence*, *checkpointing* ; tests contractuels (schémas) & *data quality gates* en CI.
- **Promotion** : *feature branches* → *PR* → *main* → *release* (env) ; *blue-green/canary* pour jobs critiques.
- **MLOps** : *registry*, *model approval*, tests de performance/robustesse, *shadow* avant *switch*.
- **Artefacts** : *Infra as Code* + *Pipeline as Code* + *Policy as Code*.

5) Checklist Architecture & Sécurité (exécutable)

5.1 Fondations

□ *Subscriptions/environnements* créés (DEV/TEST/PPRD/PROD) avec **RBAC** minimal

□ **Hub-&-Spoke** opérationnel, **Private Endpoints** pour PaaS
□ **Key Vault** / **KMS** configurés (CMK, rotation <90 j)
□ **Audit logs** routés vers SIEM/Log Analytics (rétention conforme)
□ **Naming/Tagging** & **quotas** standardisés

5.2 Données & Traitements
□ Lake *Bronze/Silver/Gold* créé (politiques lifecycle, versioning)
□ Pipelines ingestion **idempotents**, *schema validation*, *DQ rules* (≥ 90 % OK)
□ Entrepôt/lakehouse provisionné, **RLS/CLS** appliqués sur *Gold*

5.3 Gouvernance & Catalog
□ **Glossaire métier** et **catalog** en place (auto-scan, lineage)
□ **Classification PII** + politiques d'accès (ABAC par domaine)
□ **RACI** gouvernance data validé, comités calés (mensuel/trim.)

5.4 Sécurité & Conformité
□ **Zero public** (deny policies, tests d'exposition)
□ **DLP** et **exfiltration controls** actifs
□ **Tests de reprise** (RTO/RPO), sauvegardes immutables vérifiées

5.5 CI/CD & IaC
□ **Terraform/Bicep** pour infra + *pipelines* validés par *pull request*
□ **Scans IaC** & **secret scanning** *blocking*
□ **Promotion contrôlée** (gates qualité/sécurité)

6) Runbook de déploiement (pas-à-pas)

1. **Initialiser** le *repo IaC* (modules réseau, stockage, sécurité) + *workspaces* par environnement.
2. **Déployer** la **Landing Zone** (hub réseau, Key Vault, SIEM sink) → tests connectivité/deny.
3. **Créer** le **Data Lake** (containers par zone, lifecycle, encryption CMK).
4. **Provisionner** l'**orchestrateur** et le **compute** (Spark/SQL warehouses) ; intégrer l'IAM (managed identities).
5. **Activer** la **télémetrie** (metrics/diagnostics) et les **alertes** (SLO).
6. **Installer** le **Catalog** (scan des sources, lineage) + **classifier** les PII.
7. **Déployer** les **pipelines d'ingestion** *raw* (tests de schéma, DQ baseline).
8. **Configurer RLS/CLS/masquage** sur *Gold* + *policies* de partage/exports.
9. **Brancher BI & APIs** (APIM, mTLS) ; tester DLP/exfiltration.
10. **Exécuter** les **tests DRP** (restore, RTO/RPO) puis **valider** en COPIL de *go-live*.

7) KPIs/OKRs mesurables (Architecture & Sécurité)

- **IaC** : ≥ **95%** des ressources gérées par code ; *drift* = 0 (contrôles hebdo)

- **Réseau** : **0** service exposé publiquement (scans continus), 100 % PaaS via Private Link
- **Gouvernance** : ≥ **90%** des datasets **catalogués** et **classifiés** ; lineage pour ≥ 80 % des flux critiques
- **Sécurité** : rotation des secrets < **90 j** ; **100%** des accès privilégiés via **PIM** ; RLS/CLS sur **100%** des tables *Gold* contenant PII
- **Qualité** : score DQ **≥ 90%** sur *Silver*/*Gold* ; taux d'échec pipeline < **2%**
- **FinOps** : écart budget mensuel < **±10%** ; *idle time* compute < **5%**

8) Risques clés & parades

- **Dérives d'accès** → PIM, recertification trimestrielle, alertes *excess permissions*
- **Silos non gouvernés** → *Data by design*, RACI & *demand management* (processus formalisé)
- **Coûts compute** → *auto-suspend*/*auto-scale*, tagging fin/pod, budgets & alertes
- **Qualité fluctuante** → DQ *gates* en CI, contrats de schéma, *quarantine* datasets

9) Livrables finaux attendus

1. **Dossier d'Architecture Technique** (diagrammes, flux, RTO/RPO)
2. **Paquet IaC** *reproductible* + pipelines **CI/CD** (policy as code)
3. **Registre des Actifs Data** (zones, jeux, SLA) + **catalog** initialisé
4. **Plan de Sécurité** (RBAC/ABAC, RLS/CLS, DLP, clés) + **runbooks** d'incident
5. **Cockpit d'observabilité & FinOps** (SLO, alertes, budgets)