**Plataforma Unificada do Estado  
Decisão Técnica (TAD) e Estrutura Tecnológica**

Modelo global de arquitectura — simples por desenho, segura por defeito

Versão 1.0 • Data: 06/11/2025

# Índice

Nota: após abrir no Word, use “Actualizar Índice” para gerar a paginação.

# 1. Contexto e Objectivos

Este documento define as decisões técnicas fundacionais para a Plataforma Unificada do Estado (PUE): uma arquitectura browser-first, com servidores Linux Ubuntu LTS, software open‑source por defeito, dados soberanos em PostgreSQL e operação disciplinada por GitOps. O objectivo é reduzir entropia, unificar a experiência (um portal, um gateway, um catálogo de APIs) e elevar segurança, disponibilidade e velocidade de entrega.

A PUE assenta em dois data centers governamentais activos (Lisboa e Porto) com replicação em tempo real, suportando tolerância a falhas de rede e de aplicações, e garantindo continuidade de negócio com RTO/RPO definidos por camada.

# 2. Princípios e Não‑Objectivos

## 2.1 Princípios

• Browser-first e PWA para cidadania digital universal.

• OSS-first: evitar lock-in; priorizar padrões abertos (OpenAPI, OIDC, SAML, eIDAS).

• Simplicidade radical: um portal, um gateway, um catálogo de APIs. Sem cascatas de portais/gateways.

• Modular Monolith por domínio: limites claros, menos componentes, maior robustez.

• Zero Trust por defeito: tudo autenticado, autorizado e encriptado (mTLS).

• Observabilidade total e FinOps para gerir desempenho e custo por serviço.

• Acessibilidade e ética: WCAG 2.2 AA e privacidade (RGPD) desde a concepção.

• Infra como Código e GitOps: reprodutibilidade, auditoria e segurança na mudança.

## 2.2 Não‑Objectivos

• Não criar ESBs pesados ou múltiplos portais setoriais.

• Não promover micro‑serviços indiscriminadamente; usar apenas onde o custo/benefício o justifique.

• Não introduzir tecnologias fora do catálogo sem ADR aprovada.

# 3. Decisões Tecnológicas (Resumo)

• Portal Único: Front‑end React+TypeScript com Design System do Estado; PWA; WCAG 2.2 AA.

• Gateway Único: Envoy ou Traefik com mTLS, WAF (OWASP CRS), rate‑limit e logging unificado.

• Identidade: Keycloak (OIDC/SAML), MFA/WebAuthn, integração eIDAS/Cartão de Cidadão.

• Back‑end: Padrões: FastAPI (Python) e Spring Boot (Java); APIs OpenAPI; gRPC interno quando necessário.

• Dados OLTP: PostgreSQL 16+ com Patroni (HA). Citus/Timescale conforme necessidade de escala ou séries temporais.

• Objectos: Ceph (S3 RGW / RBD) com snapshots e política WORM para backups imutáveis.

• Analítica: ClickHouse para consultas rápidas; Apache Superset para BI; catálogo Apache Atlas e qualidade com Great Expectations.

• Eventos: NATS para eventos/filas; Debezium (CDC) quando a integração for via logs de BD.

• Kubernetes: Clusters upstream sobre Ubuntu + containerd, um por DC; GitOps (Argo CD); OPA/Gatekeeper para políticas.

• Segurança: Wazuh, Suricata, Zeek, Velociraptor; gestão de segredos com OpenBao (Vault OSS) + HSM; assinatura de imagens (cosign).

# 4. Estrutura Tecnológica

Matriz de camadas com tecnologias padrão, alternativas aprovadas e observações operacionais.

|  |  |  |  |
| --- | --- | --- | --- |
| Camada | Padrão (Tecnologia) | Alternativas Aprovadas | Observações |
| Experiência (Web/PWA) | React+TS; Design System; WCAG 2.2 | Vue (casos justificados) | Um portal único com workspaces por perfil. |
| Gateway de Entrada | Envoy/Traefik; mTLS; WAF (OWASP CRS) | — | Sem cascatas de gateways; logs centralizados. |
| Identidade e AAM | Keycloak (OIDC/SAML); MFA/WebAuthn; eIDAS/CC | — | Sessões curtas; políticas de acesso por papel; mapeamento de atributos. |
| Back‑end | FastAPI (Python) / Spring Boot (Java) | gRPC interno | Modular monolith por domínio; OpenAPI obrigatório. |
| Eventos & Integração | NATS (pub/sub, filas) | Debezium (CDC) | Evitar ESBs pesados; integração leve e observável. |
| Dados OLTP | PostgreSQL 16+ (Patroni) | Citus; TimescaleDB | HA com quórum; índices e partições por workload. |
| Object Storage | Ceph (S3 RGW / RBD) | — | Snapshots; políticas WORM; versionamento. |
| Analítica/BI | ClickHouse; Superset | — | Catálogo de dados (Atlas) e qualidade (GE) obrigatórios. |
| Plataforma | Kubernetes (Ubuntu + containerd) | OpenStack para IaaS | GitOps (Argo CD); políticas OPA; cosign em imagens. |
| Observabilidade | Prometheus, Loki, Tempo, Grafana | OpenTelemetry | Traços fim‑a‑fim; SLO por serviço. |
| Segurança | Wazuh; Suricata; Zeek; Velociraptor | RASP quando necessário | Detecção de anomalias; playbooks de resposta. |
| Segredos | OpenBao (Vault OSS) + HSM | — | Rotação automática e mTLS com SPIFFE/SPIRE (quando adoptado). |
| CI/CD | GitLab CE; Harbor; Argo CD | — | Pipelines SAST/DAST; assinatura e policy enforcement. |
| IaC | Terraform; Ansible | — | Reprodutibilidade e auditoria integral. |

# 5. Alta Disponibilidade, DR e Metas de Serviço

Dois data centers activos (Lisboa e Porto). Replicação PostgreSQL síncrona intra‑DC e síncrona/quase‑síncrona inter‑DC conforme latência. Ceph multi‑site (RBD mirroring / S3 multi‑region). Witness governamental para quórum.

|  |  |  |  |
| --- | --- | --- | --- |
| Camada | Disponibilidade | RTO (máx.) | RPO (máx.) |
| Identidade / DNS / Rede | ≥ 99,99% | 5 min | 0–30 s |
| Transaccionais (OLTP) | ≥ 99,95% | 15 min | < 60 s |
| Analítica / BI | ≥ 99,9% | 60 min | ≤ 5 min |

# 6. Segurança (Zero Trust)

• mTLS entre serviços; identidade de workload (SPIFFE/SPIRE quando aplicável).

• Gestão de segredos com HSM; rotação automática; segregação de funções.

• SIEM/EDR: Wazuh, Suricata, Zeek, Velociraptor; uso de IA para detecção de anomalias.

• WAF no edge; RASP selectivo para aplicações críticas; baselines CIS para Ubuntu/Kubernetes.

• RGPD: classificação de dados, pseudonimização, registos de auditoria imutáveis (WORM).

# 7. Governação de APIs e Dados

• Catálogo único de APIs (OpenAPI/AsyncAPI) — publicação obrigatória antes de ir a produção.

• Versionamento: SemVer; APIs versionadas por cabeçalho ou caminho; depreciação com calendário.

• Catálogo de dados (Apache Atlas); qualidade com Great Expectations; contratos de dados por domínio.

• Partilha por eventos (NATS) e por CDC (Debezium) quando apropriado; proibida a ligação directa a BD de outro domínio.

# 8. Padrões de Entrega e Operação

• CI/CD em GitLab: SAST/DAST, testes, assinatura de imagens (cosign), promoção por ambiente.

• GitOps (Argo CD): todo o estado declarativo; \*drift detection\* e auditoria de mudanças.

• Observabilidade obrigatória: métricas, logs e traços; SLO por serviço e \*error budget\*.

• FinOps: custo por serviço/domínio; \*showback/chargeback\* por ministério; metas de eficiência.

• Exercícios de DR trimestrais com relatório público dos resultados e acções correctivas.

# 9. Fases de Implantação

|  |  |  |
| --- | --- | --- |
| Fase | Conteúdo | Janela |
| 0 — Fundações | Baselines de segurança; Keycloak; gateway único; GitLab/Harbor; observabilidade; padrões UX. | 0–3 meses |
| 1 — Núcleo de Dados & DR | PostgreSQL/Patroni; Ceph multi‑site; backups imutáveis; clusters K8s Lx/Pt. | 3–6 meses |
| 2 — PUE & Pilotos | Portal Único; 2–3 domínios piloto (agendamento, pagamentos, dossiê do cidadão). | 6–12 meses |
| 3 — Escala | Expansão por domínios; desligar legados; analítica ClickHouse; IA de segurança. | 12–24 meses |
| 4 — Optimização Contínua | SLA/SLO por serviço; DR drills; governação de dados madura. | 24+ meses |

# 10. Métricas de Sucesso

• Simplicidade: portais reduzidos a 1; gateways a 1; catálogos duplicados = 0.

• Qualidade: erros 5xx < 0,1%; \*lead time\* de \*deploy\* < 1 dia; MTTR < 30 min.

• Resiliência: uptime ≥ 99,95%; RPO crítico ≤ 30 s.

• Segurança: CVEs críticas corrigidas < 7 dias; \*phishing click-rate\* ↓ 70%.

• Custos: custo por transacção decrescente (FinOps).

# 11. Riscos & Mitigações

• Resistência organizacional → mitigação: patrocínio executivo e incentivos de consolidação.

• Déficit de competências K8s/OSS → mitigação: academia interna, \*pairing\* com equipas piloto.

• Latência inter‑DC → mitigação: afinação de quórum e \*placement\*; usar quase‑síncrona quando necessário.

• Entropia tecnológica → mitigação: catálogo estrito e ADRs obrigatórias.

# 12. Registos de Decisão (ADR) — Resumo

ADR‑001 — Um Portal Único do Estado (PUE) — Aprovado.

ADR‑002 — Gateway Único (Envoy/Traefik) — Aprovado.

ADR‑003 — Identidade com Keycloak (OIDC/SAML) — Aprovado.

ADR‑004 — Back‑end padrão FastAPI/Spring — Aprovado.

ADR‑005 — Dados em PostgreSQL/Patroni — Aprovado.

ADR‑006 — Object storage com Ceph — Aprovado.

ADR‑007 — Analítica com ClickHouse/Superset — Aprovado.

ADR‑008 — Eventos NATS + CDC Debezium — Aprovado.

ADR‑009 — K8s (Ubuntu+containerd) + GitOps — Aprovado.

# 13. Figuras

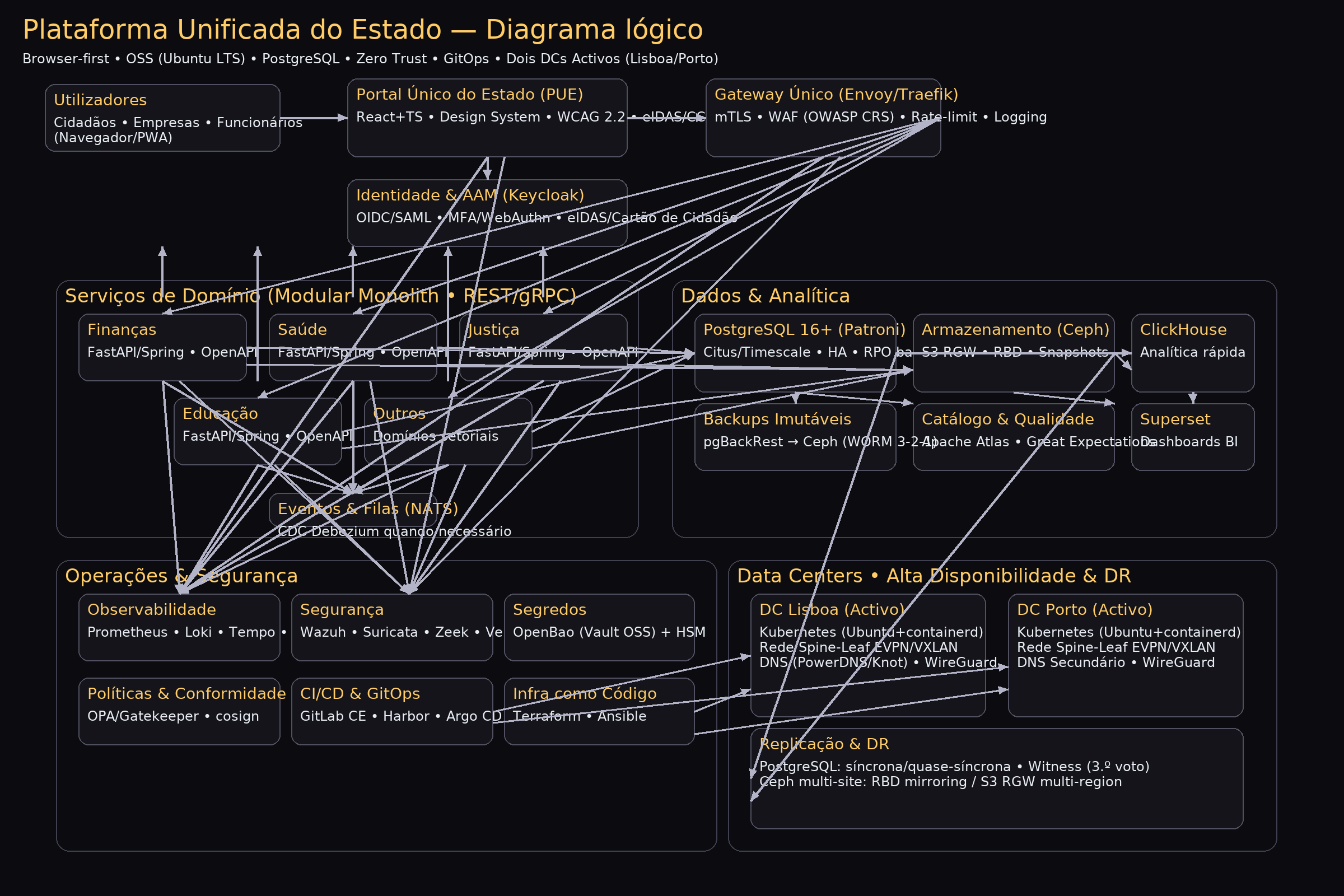


Figura 1 — Diagrama lógico (completo)

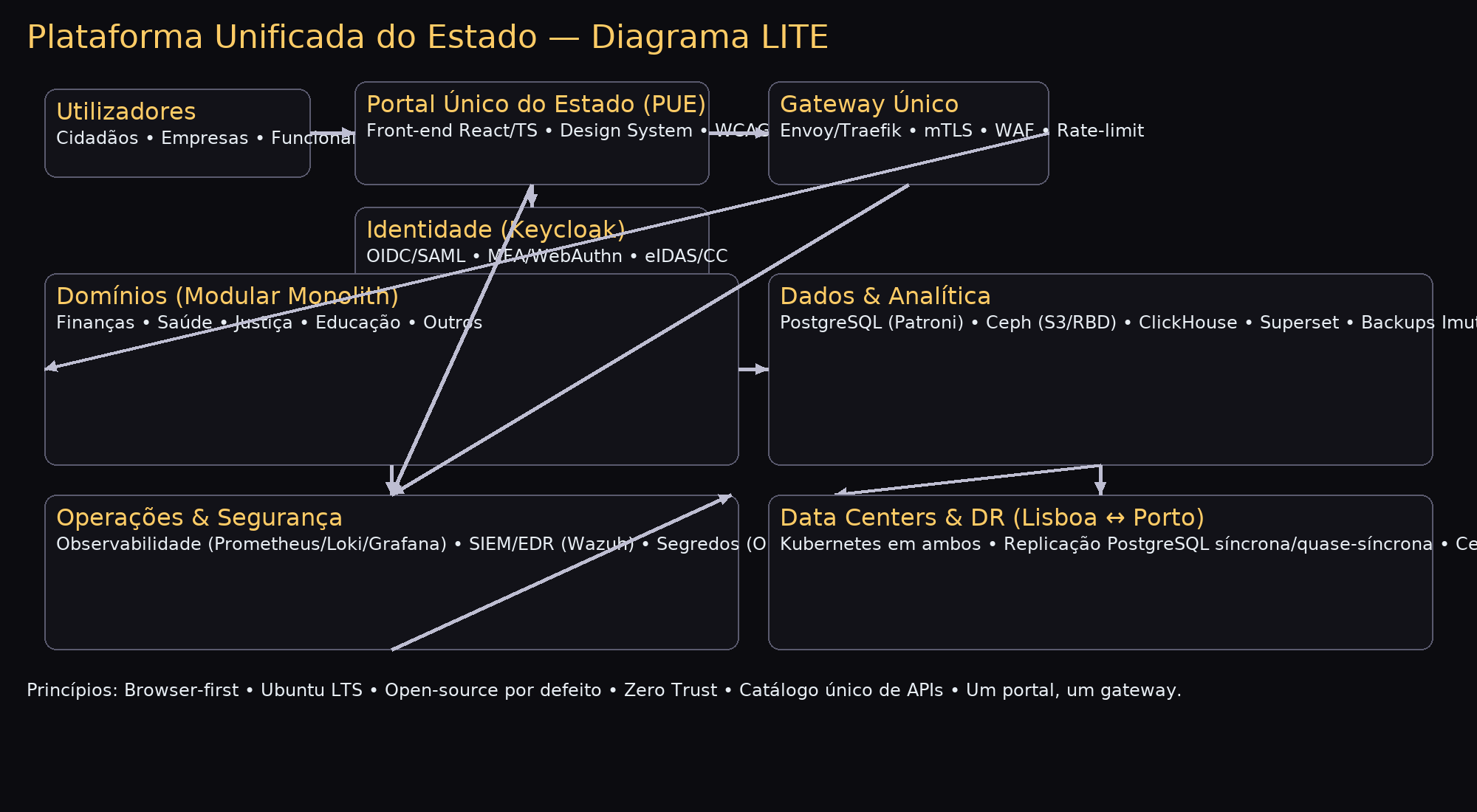


Figura 2 — Diagrama lógico (LITE)

# 14. Anexos

A. Baselines CIS (Ubuntu/Kubernetes) — lista de controlo resumida.

B. Modelos de SLO e \*error budgets\*.

C. Templates de ADR.

D. Playbooks de resposta a incidentes (IR).