

Perfeito! Aqui vai a **justificativa, ponto a ponto**, de tudo que propus no **Tópico 2 – Arquitetura Macro**. A ideia é você poder “defender” cada escolha em banca técnica (qualidade, segurança, compliance, escalabilidade e ROI).

---

## 1) Canais (Web e Mobile) + MFA/biometria

### Por quê?

- **Next.js (Web):** SSR/ISR melhora performance e SEO do catálogo de pitches; DX madura, enorme base de talentos.
  - **React Native/Expo (Mobile):** ciclo de release rápido, reuso de componentes entre web/mobile, OTA updates quando cabível.
  - **MFA + biometria:** reduz fraude e se alinha ao seu requisito de **validação de identidade** em eventos críticos (fim do cadastro, publicar pitch, aprovar empréstimo).  
**Valor:** menor atrito de desenvolvimento, **time-to-market** rápido e segurança de ponta no canal onde a fraude acontece (cliente).
- 

## 2) BFF/Edge (GraphQL + WAF + rate limiting + idempotência)

### Por quê?

- **BFF** centraliza regras de agregação e evita acoplamento dos apps com microservices, reduzindo retrabalho.
  - **GraphQL** elimina over/under-fetching; útil num **feed com filtros complexos** (catálogo).
  - **WAF/Rate-limit/Bot mgmt:** proteção na borda para endpoints sensíveis (login, originação, Pix).
  - **Idempotência:** evita transações duplicadas (ex.: clique duplo em “contratar”), crítico em pagamentos.  
**Valor:** performance previsível, **segurança na borda**, APIs estáveis para os apps e **operacionalidade** em pagamentos.
-

### 3) Domínios (microservices por contexto de negócio)

#### Por quê?

Separar em: Onboarding/KYC, Antifraude, Score & Risk, Catálogo/Suitability, Originação & Pricing, Pagamentos/Carteira, Contratos, Servicing/Cobrança, Compliance.

- **Isolamento regulatório** (PII e AML/COAF);
- **Escalabilidade independente** (picos em catálogo ≠ picos em pagamentos);
- **Time ownership** por domínio;
- **Mudanças seguras** sem “ripple effects”.

**Valor:** arquitetura evolutiva e **resiliente**, pronta para crescer sem reescrever o core.

---

### 4) Dados & Integrações (Postgres, Redis, Kafka, OpenSearch, Lake/DW, MLOps)

#### Por quê?

- **PostgreSQL:** OLTP transacional robusto (ACID) e features financeiras (precisão decimal).
- **Redis:** cache/locks/sessões para latências baixas e controle de concorrência.
- **Kafka:** backbone de eventos (LoanCreated, KYCVerified, PaymentSettled) → **auditabilidade** e **acoplamento fraco**.
- **OpenSearch:** busca/filtros do **feed** de pitches (texto, ranges, ordenações).
- **Data Lake + DW/BI:** relatórios regulatórios, KPIs (inadimplência, LGD, coortes), modelagem de risco com **dados históricos**.
- **Feature Store/Registry (MLOps):** garante reprodutibilidade e **consistência** entre treino e produção.

**Valor:** dados governados, **explainability** e **compliance** sem travar performance do produto.

---

## 5) Plataforma & SRE (EKS, mesh mTLS, observabilidade, CI/CD, Vault/KMS, DR)

### Por quê?

- **Kubernetes (EKS):** padroniza deploy/escala, isolates por namespace/ambiente.
  - **Service mesh (mTLS):** zero-trust interno, telemetria e políticas de tráfego.
  - **Observabilidade (OTel/Prom/Grafana/ELK):** métricas, logs e traces → MTTR baixo e RCA confiável.
  - **CI/CD com gates de segurança** (SAST/DAST, SBOM, assinatura de imagem): supply-chain security.
  - **Vault + KMS/HSM:** gestão de segredos e chaves; rotação automática.
  - **Backups & DR:** RPO/RTO definidos (multi-AZ e plano cross-region).  
**Valor:** confiabilidade e **segurança operacional** nível enterprise.
- 

## 6) Justificativas por domínio

### 6.1 Onboarding & KYC/KYB

#### Por quê?

- Cumpre **KYC/AML** (PEP, sanções) e reduz **fraude de identidade**.
  - **Step-up biométrico** nos seus três pontos críticos limita “account takeovers”.
- Valor:** base de clientes limpa, **risco reduzido** e aderência regulatória.

### 6.2 Antifraude & Device Risk

#### Por quê?

- Fraude migra para **conta/dispositivo**; device fingerprint + velocity rules capturam anomalias.
  - **Modelos de anomalia** (Isolation Forest/GBMs) reduzem falsos positivos e automatizam escalonamentos.
- Valor:** menos chargeback/fraude, sem penalizar usuários bons.

### 6.3 Scorecredit & Risk Engine

#### Por quê?

- Une **Open Finance + Serasa + histórico interno** → visão 360 do risco.
- **Explainability (scorecard/SHAP)**: decisões auditáveis (exigência de risco/Compliance).
- **Challenger models/monitor de drift** mantêm performance ao longo do tempo.  
**Valor**: precificação justa (**CDI + spread 5/10/15**), menor inadimplência e confiança para investidores.

### 6.4 Catálogo / Matchmaking / Suitability

#### Por quê?

- **Ranqueamento multi-sinal** (risco, retorno, prazo, concentração) maximiza match “bom para ambos”.
- **Suitability** (perfil do investidor) evita alocação indevida e **melhora retenção**.
- **Busca/filtros** (OpenSearch) dão **controle** ao investidor.  
**Valor**: engajamento alto e **conversão** com segurança.

### 6.5 Originação & Pricing

#### Por quê?

- **CDI + spread por faixa (A/B/C = +5/+10/+15)** é linguagem de mercado e acompanha o ciclo.
- **Convenção 252 d.u.** e simuladores (SAC/PRICE) dão resultado **auditável**.
- **Idempotência** impede contratos “duplicados” sob falhas de rede.  
**Valor**: taxas transparentes, **defesa técnica** e menos incidentes operacionais.

### 6.6 Pagamentos & Carteira (Pix) com razão de dupla entrada

#### Por quê?

- **Double-entry ledger** garante integridade contábil (débito = crédito).
- **Webhooks + conciliação** asseguram que o Pix liquida e contabiliza corretamente.  
**Valor**: confiança contábil e **reconciliação** simples em auditorias.

## 6.7 Contratos & Assinatura Eletrônica

### Por quê?

- **Trilha de assinatura** (IP, device, timestamp) e **WORM storage** → **não-repúdio**.
- **Hash do PDF** e carimbo temporal facilitam auditorias e (futuro) **ancoragem blockchain**.  
**Valor:** segurança jurídica dos acordos P2P.

## 6.8 Servicing, Cobrança & Recuperação

### Por quê?

- **D+1** repasse aumenta confiança do investidor.
- **Servicing fee 0,5% a.a.** financia gestão, suporte, cobrança e relatórios.
- **Orquestrador** (agendas, reprocessos) previne erros e atrasos.  
**Valor:** receita **recorrente** + qualidade de operação (menos churn, mais reinvestimento).

## 6.9 Relatórios & Compliance

### Por quê?

- **AML/COAF**, limites transacionais, reconciliações diárias e retenção de logs são obrigatórios para operar com parceiros regulados.  
**Valor:** evita penalidades, dá previsibilidade a auditorias e **abre portas** com parceiros (ex.: SEP/SCD/BaaS).

---

## 7) Segurança, Privacidade e Auditoria (LGPD, zero-trust, WORM)

### Por quê?

- **LGPD:** minimização, bases separadas (PII vs. analytics), pseudonimização e bases legais por tratamento.
  - **Zero-trust** (mTLS, RBAC/ABAC): cada serviço e cada operador vê **só o necessário**.
  - **Logs imutáveis (WORM) + carimbo de tempo:** trilhas forenses confiáveis.  
**Valor:** reduz risco legal, melhora postura de segurança e acelera due diligences.
-

## 8) Blockchain (hashes agora; tokenização depois)

### Por quê?

- **Hoje:** registrar **hashes** de contratos/eventos críticos em **ledger permissionado** + **ancoragem pública** → prova de integridade **sem** custar latência/custo do core.
  - **Amanhã:** **tokenização interna** de participações e **smart contracts** para waterfall do fundo de mitigação (demanda avaliação regulatória).  
**Valor:** reputação e **verificabilidade** crescentes sem travar o produto.
- 

## 9) Catálogo/Feed (UX + transparência)

### Por quê?

- **Ranking** por risco/retorno/prazo e filtros claros → o investidor entende “por que este pitch”.
  - **Explicabilidade** no card (ex.: fatores que geraram risco B) aumenta a **confiança**.
  - **Controles de concentração** evitam “all-in” num único tomador/setor.  
**Valor:** conversão com educação financeira embutida e **gestão de risco** melhor.
- 

## 10) Dados & Analytics (lakehouse + DW/BI + governança de modelos)

### Por quê?

- **Lakehouse** centraliza históricos para risco e fraude; **DW/BI** entrega visão executiva e regulatória.
  - **Governança de modelos** (versionamento, métricas, drift) evita “caixa-preta” e mantém performance.  
**Valor:** decisões orientadas a dados e **auditáveis**.
-

## 11) Operação & Resiliência (SLOs, chaos, feature flags, runbooks)

### Por quê?

- **SLOs por domínio** (p95, uptime) alinham produto/engenharia com o que importa ao usuário.
- **Chaos testing e feature flags** evitam quedas amplas e permitem rollouts seguros.
- **Runbooks e post-mortems** padronizam resposta a incidentes e aprendizado.

**Valor: previsibilidade** operacional e melhoria contínua.

---

### Observação rápida sobre Origination Fee

Mantivemos **2% do tomador** (como alinhamos) para equilibrar **conversão e receita**. Se vocês preferirem **10%**, é defensável, mas: aumenta atrito de contratação, eleva risco de **seleção adversa** (bons tomadores fogem) e pode reduzir recorrência de uso. O framework acima funciona com qualquer percentual — a escolha é estratégica.