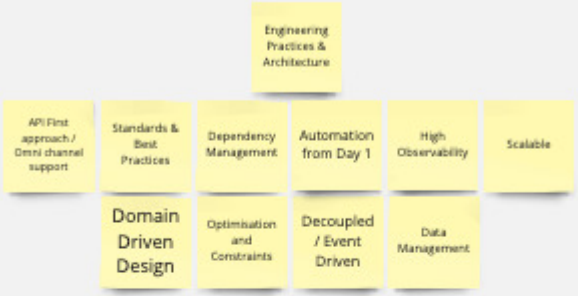
Chaiwala Engineered

1. Technology Management
2. People Management
3. Product Planning & Delivery
4. Engineering Quality
5. Application Software Architecture
6. Systems Design Approach
7. Software Design Artefacts
8. API Design
9. Design for scale
10. Deployment Safety

Technology Management



People Management

People Management

Roles & Personas

Org / Team Structures

Culture

Environment

Recruitment & Hiring for the Long Term

Levels of focus & responsibility

Autonomy

Coaching / Mentoring

Values, Tenets & Principals

Goal setting

Performance Management

Profiles

Promotions

Coaching plans

Training

1:1 Feedback - Weekly, Quarterly, Bi-Annually

Reporting - Upstream, HR

History

Outlook

Peer Feedback

High Impact Areas

Growth Areas

Strengths

Drive Continuous Improvement

Actively Share the Bigger Picture and individual role in it

Regular discussions - weekly/monthly/quarterly goals, suggest goals / opportunities

Remove barriers - red tape, meetings, time-boxing

Drive learning activities - stretch goals, xTeam consults, SPOC, POCs, Mentorships, Brown bags, Hackdays

Invest in self-service & automation - mundane ops work, customer training/user guides

Solicit feedback for improving job & customer experience

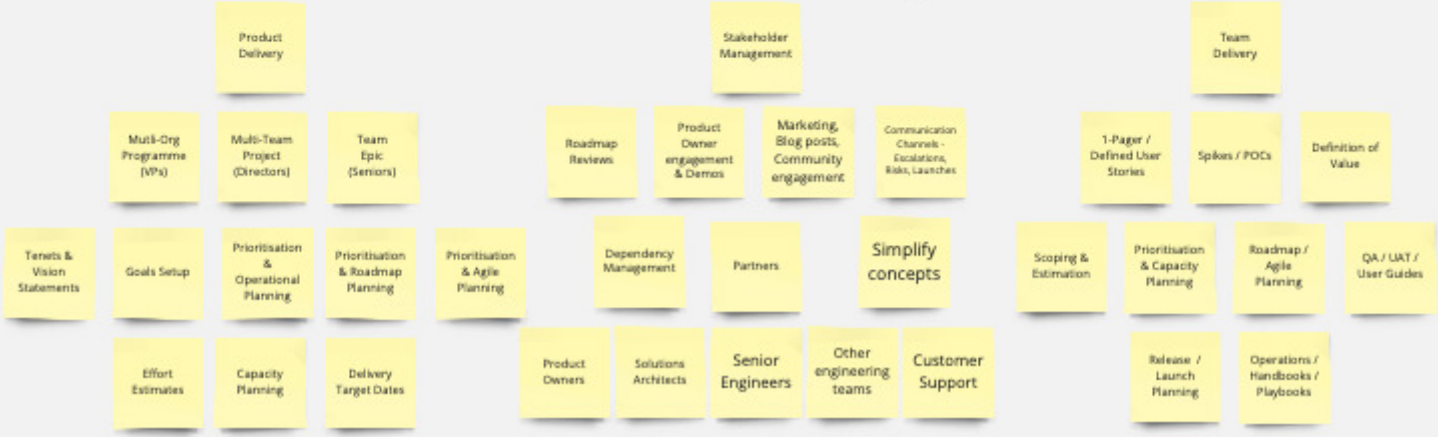
Actively solicit customer feedback and share with team - empower team to build relationship

xTeam pollination - retros, standup, roadmap reviews, tech talks, secondments

Support small ideas - generate ideas stream to improve CeX, Ops, SDLC, Meetings, Paper cuts

Product Planning

Product Delivery



Engineering Quality

Product Quality

- Consistent UX / UI
- Regulatory Compliance
- Engagement Channels
- Customer Support
- User Testing / Customer Reviews
- Market Ananlysis
- Accessibility
- L10n / I18n
- Self serviceabiltiy
- User guides
- Internal Tooling / Process

Systems Quality

- Diagnostics
- Architectural Consistency
- Runtime Dependencies
- Observability
- Availability
- Technical Debt
- Scalability
- Cost reduction
- Key metrics
- Monitors & Alarms
- Benchmarking / Load testing

Code Quality

- Valuable Commit Message
- Automated Tests coverage
- Release Notes
- Static Analysis Results - e.g. Tricorder for compiler warnings, unused/deprecated dependencies/variables etc, doc format, potential bugs
- Intended Function
- Code
- Product Owner
- Change Description - Purpose/Motivation, Doc links
- Customer Impact, Risks, Key Metrics
- Deployment Guide
- Release Dependencies - H/W and S/W
- New Build Dependencies
- Customer Impact / Key Metrics
- Linter - Security, Code smells, bugs, styling, naming standards
- Comments / Documentation
- Design & Runtime Complexity
- Consistency
- Redundancies & Tech debt
- One way doors / CeX
- Compatibility (B & F)
- Error Handling
- 2 person approval
- Iteration over big bang
- CR queue
- Exemption - class of changes, e.g. styling
- Senior vs Junior SDEs
- Time drift - Review schedules

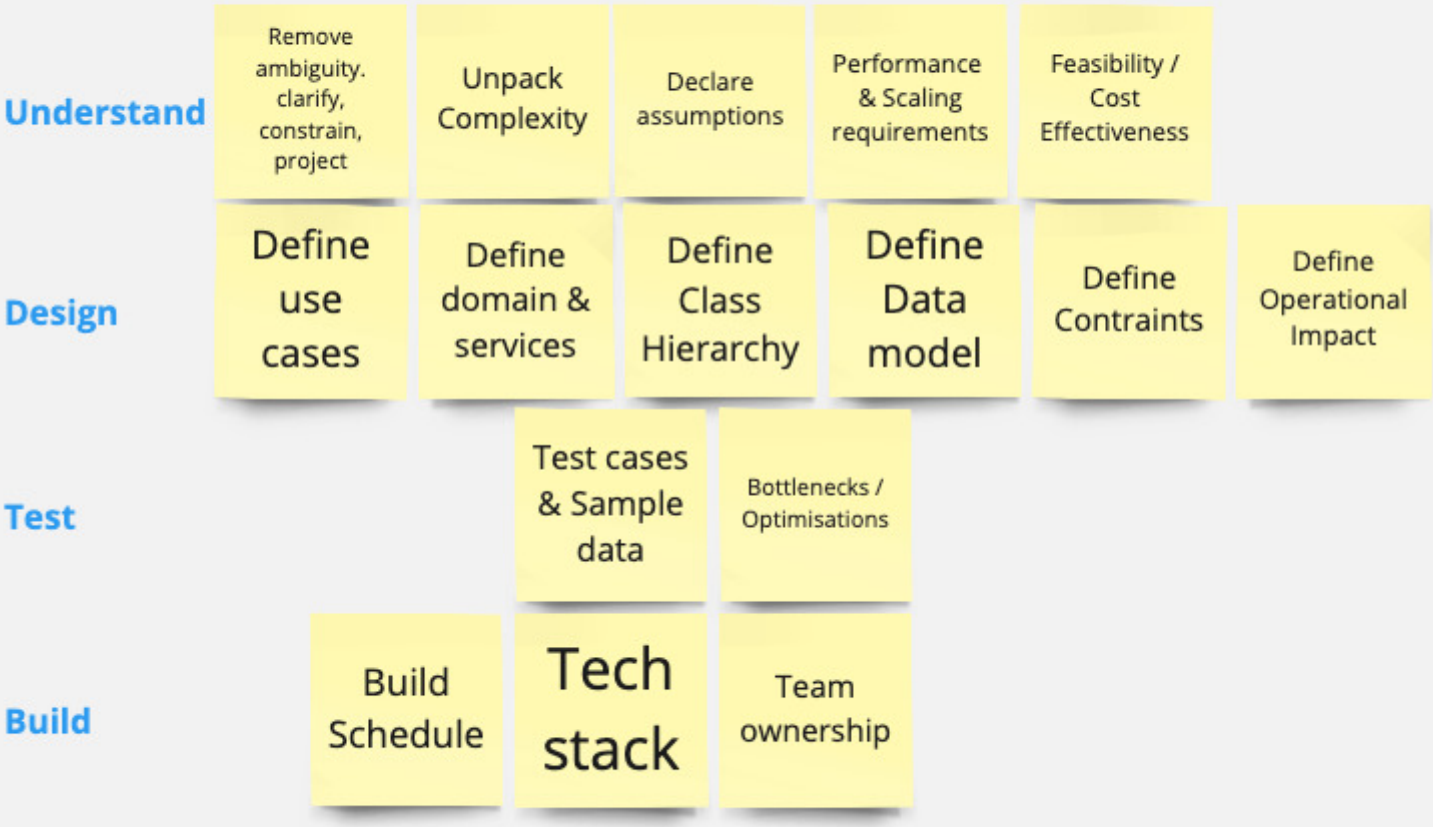
Application Architecture

- Monolith
- Hybrid
- SOA
- Microservices
- Messaging / Integration
- Releases & Deployments
- Domain Driven Design
- Cells and Clusters

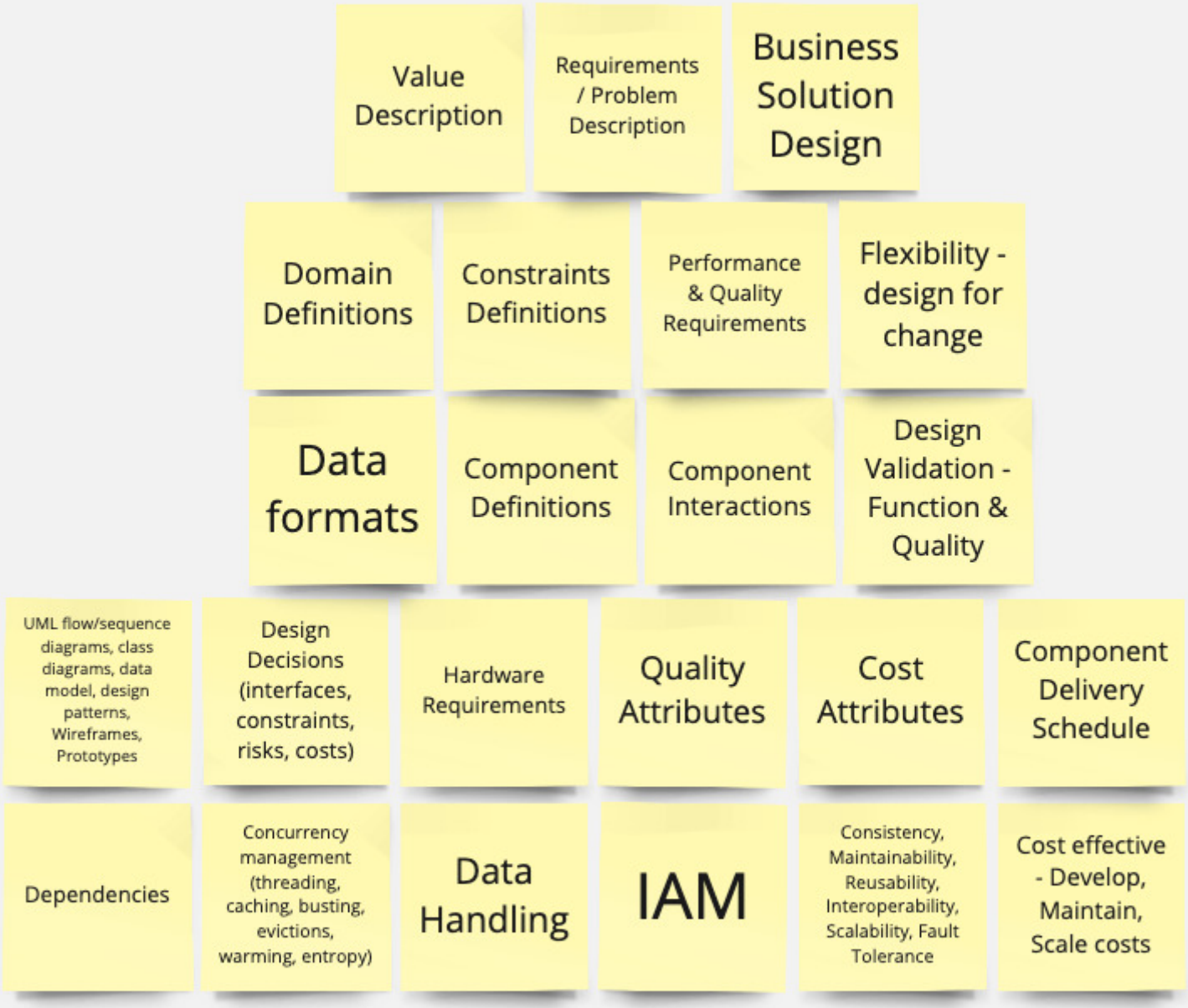
Infrastructure Architecture

- Compute
- Networking
- Data Storage & Archival
- IT Systems

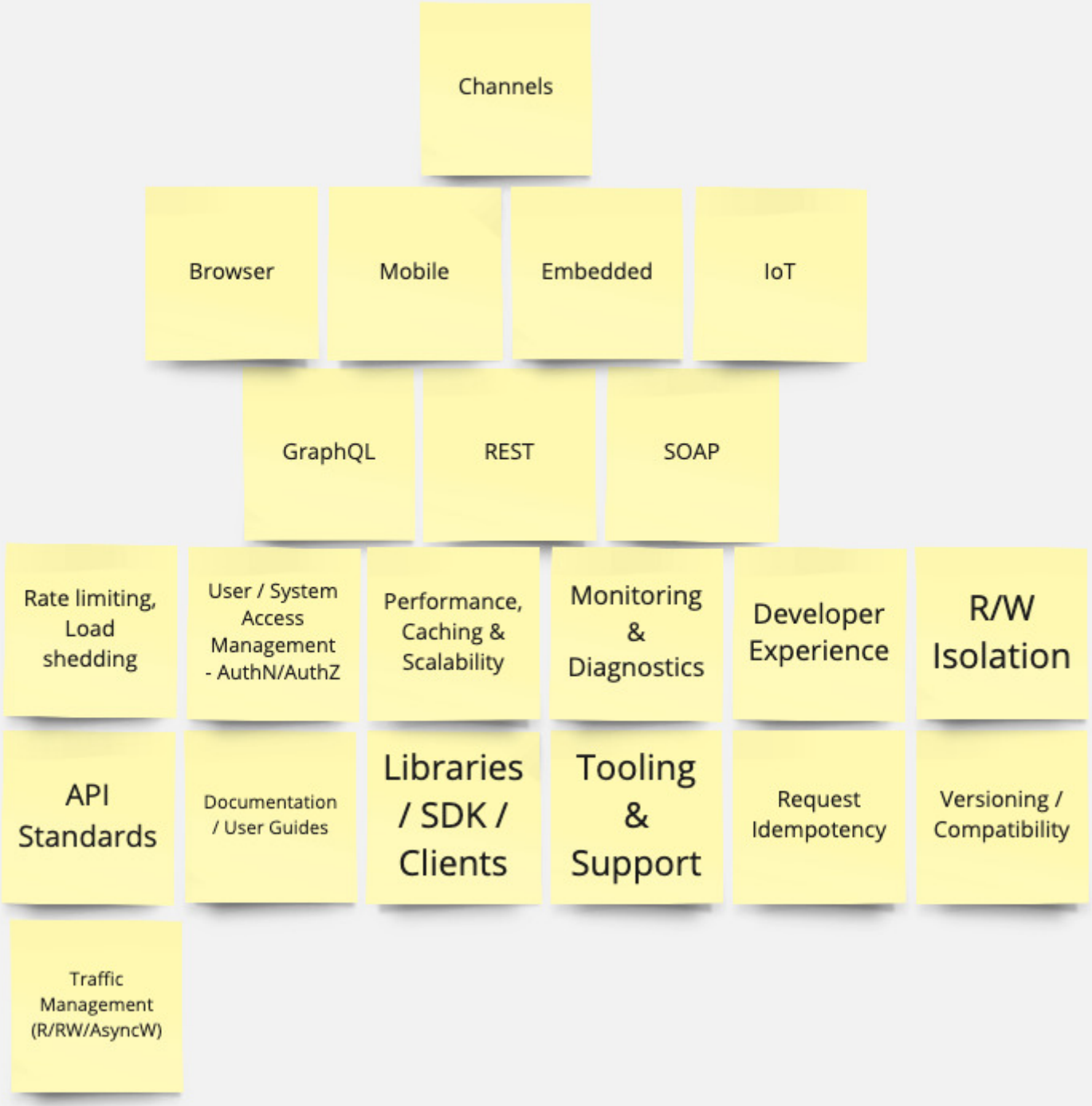
Systems Design Approach



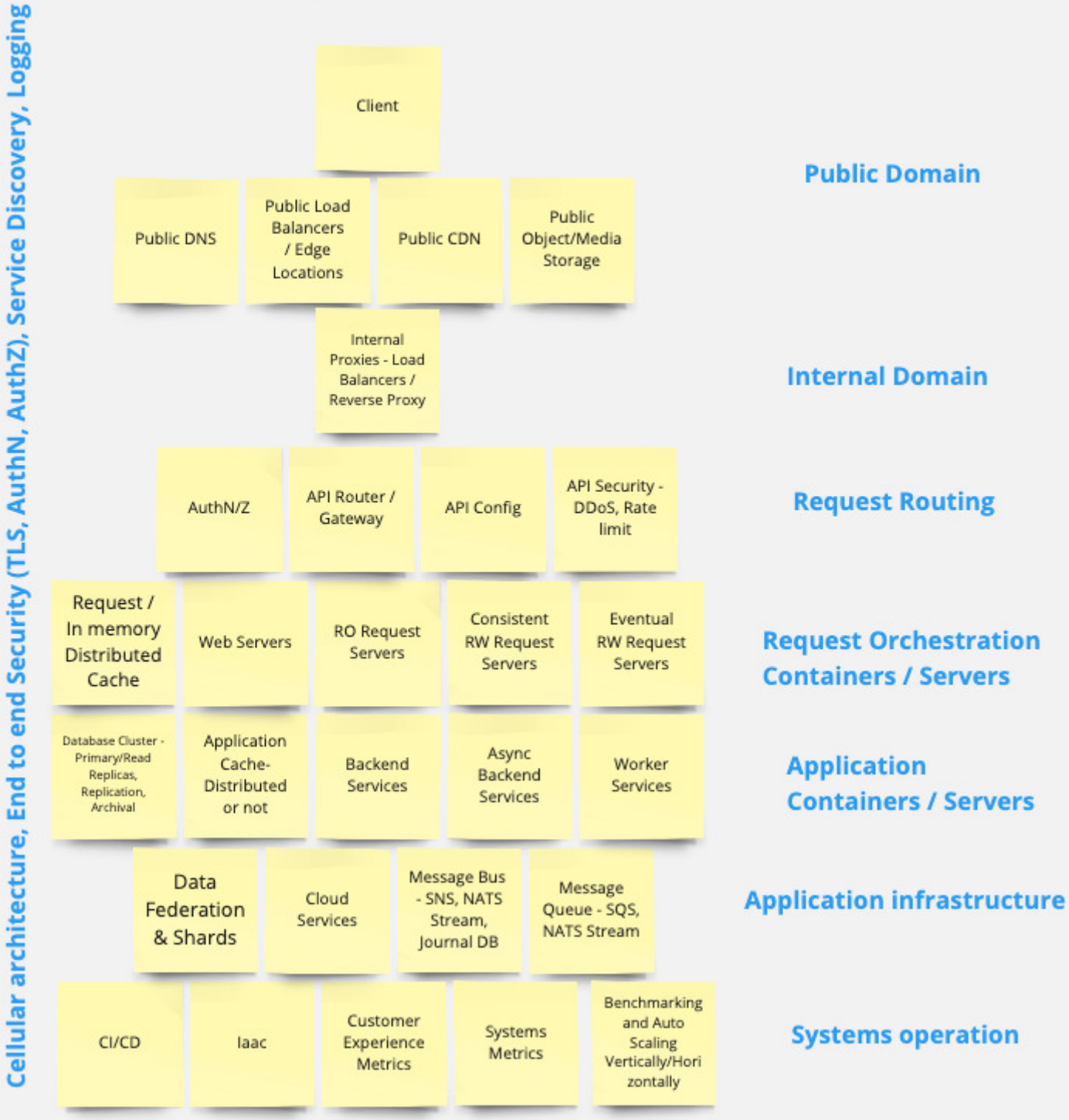
Software Design Artefacts



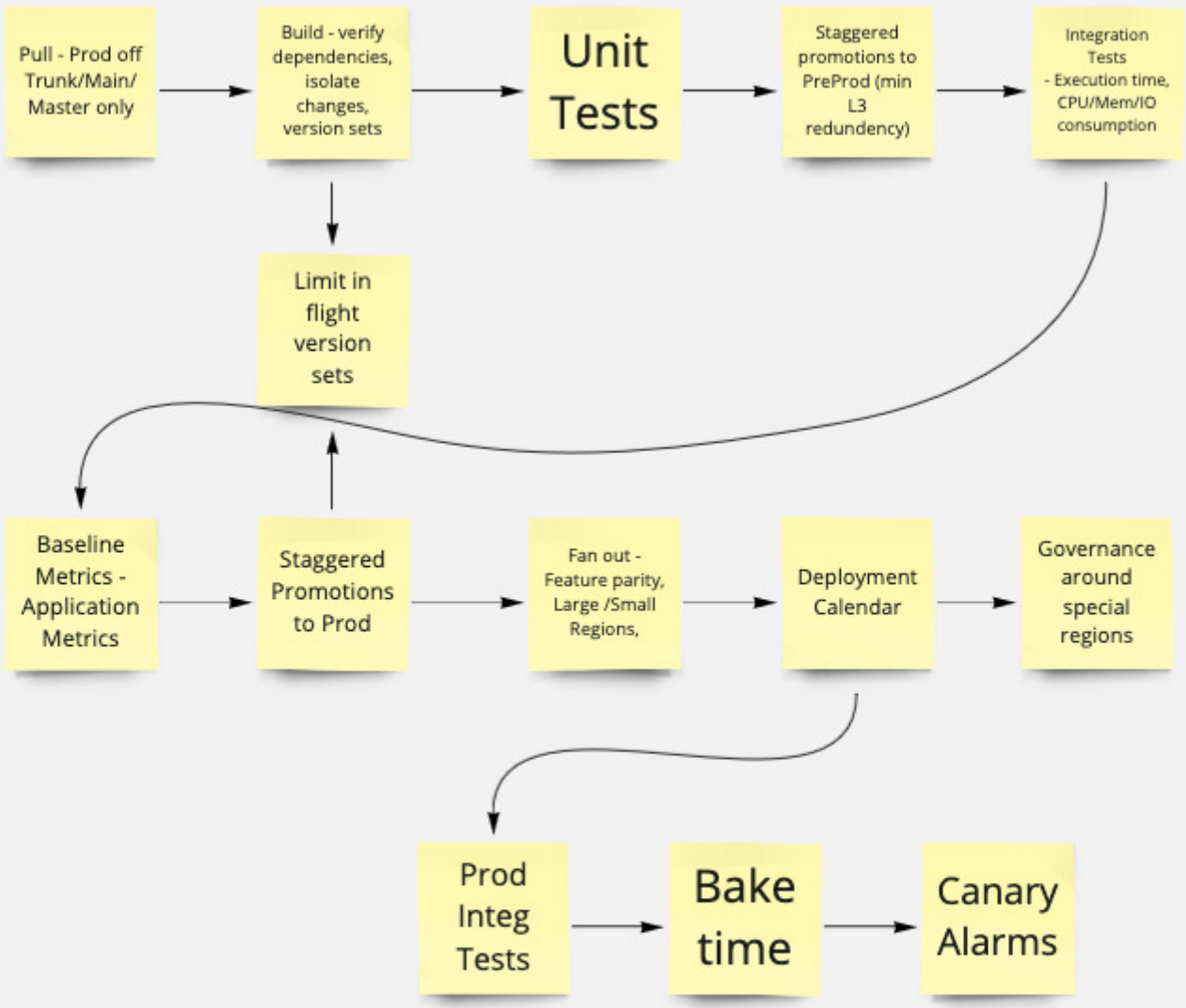
API Design


Channels
Browser
Mobile
Embedded
IoT
GraphQL
REST
SOAP
Rate limiting, Load shedding
User / System Access Management - AuthN/AuthZ
Performance, Caching & Scalability
Monitoring & Diagnostics
Developer Experience
R/W Isolation
API Standards
Documentation / User Guides
Libraries / SDK / Clients
Tooling & Support
Request Idempotency
Versioning / Compatibility
Traffic Management (R/RW/AsyncW)

Design for scale



Deployment Safety



Rollbacks

Tests, Alarms, Canaries, Emergency

On host VSet backup

Staggered - reverse order

Deployment Alarms, Emergency actions, Visualisers, Rollbacks, Global vs Regional Consistency

Oracle Values - initial thoughts

- **Put Customers First**
  - Different customers or stakeholders involved
  - Actively engage
  - Make every engagements easier
  - Connect, understand, deliver value
  - Workshops, Visits, Beta Prog, Ops metrics
  - EIP Tagging - Needed POC for Console coordination & testing
- **Act now, Iterate**
  - Deliver Value as soon as possible & safely, then iterate
  - API config - region builds, customer burden, take on manual ops
  - Feature launches - portal to generate CMs, then build unified CeX
- **Nail the basics**
  - Preempt customers
  - Automation
  - Papercuts
  - API Config
  - Deployment tracking, User guides etc
  - Reports, deployments, code review linters
  - Slow you down, Creates frustration
  - Challenge wasn't in UX but the BE
- **Expect & Embrace Change**
  - Product Changes
  - Org changes
  - Naming change (Flexible RI)
  - Restructure teams to best deliver business goals
- **Earn Trust, Give Trust**
  - Different customers or stakeholders involved - Collaborative mindset
  - Understand workloads, involve stakeholders / customers
  - Oncall rotations, Anticipate needs, communicate openly
  - Empower through effective communication - check-ins, milestones
- **Take Risks, Remain Calm**
  - Pioneering mindset
  - Use metrics
  - Expedited Deployments
  - Playbooks help to stay calm
  - Embedding SysDEs, Company level change to dedicated away teams
  - Basis of pre and post decision making
  - Align with / Support other teams deployments
  - One way doors, rollbacks, customer messaging
- **Don't be a jerk**
  - Promote togetherness
  - Be constructive & forward looking
  - Respect your past
  - Eliminate org beauracracy
  - Pulling rank, Overpowering conversations
  - Architecture & Design decisions
- **Own without Ego**
  - Chain of responsibility vs. Total isolated ownership
  - Actively disprove your beliefs/biases
  - Actively build feedback loops
- **Take pride in your work**
  - Details matter, even if only to you
  - Remember to celebrate! - this one catches my introvert bias often
  - Actively disprove your beliefs/biases