ArchDays 2022 · October 21, 2022

How to pass a System Design Interview

Preparation and candidate behavior

ARCHDAYS

Speaker

Alexander Polomodov

CTO, mobile banking and acquisition, Tinkoff

Focus

Oversaw architecture decisions

Context

Oversaw System Design Interviews · Interviewed technical leaders

From hiring to candidate behavior

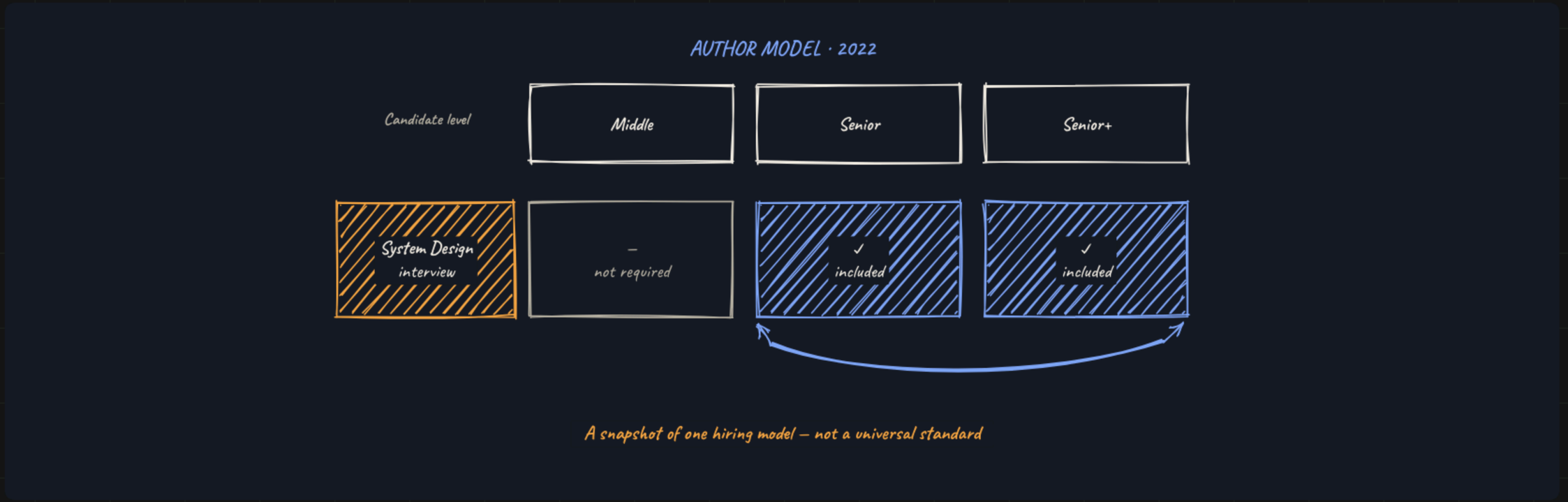
Map of the talk

- Hiring and seven steps
- Requirements, boundaries, flows
- Models, technology, scaling
- Behavior and study paths

Hiring Author's model · 2022

System Design targets Senior and Senior+

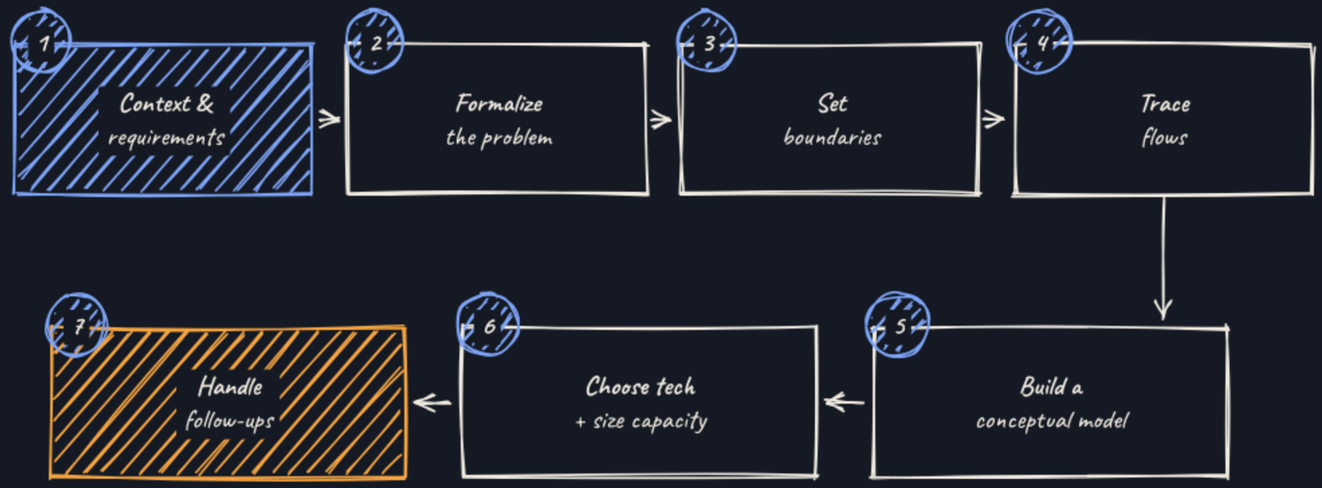
Junior and Middle skip this section



Interview 7 steps

Architecture emerges one step at a time

From the prompt to follow-up questions



Step 2

02

Formalization

Before the first line is drawn

Clarify the problem first

Scope `Ask first`

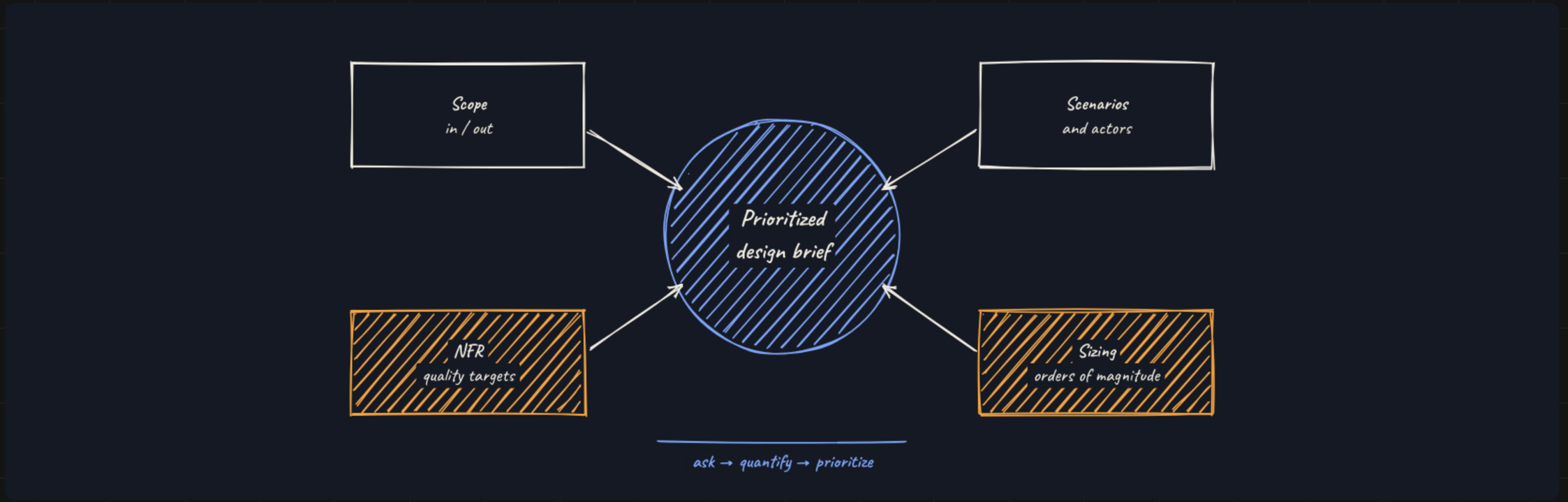
Scenarios, scale, priorities

- What is in and out
- Who uses it and why
- What volumes are expected
- What matters most

NFRs Trade-offs

Priorities direct the architecture

Sizing connects requirements to scale



Preparation Methods

Learn the language of requirements

Functionality and architecture qualities

- UML Use Cases
- User Stories and JTBD
- ATAM for trade-offs
- Wiegers and Eyskens

Step 3

03

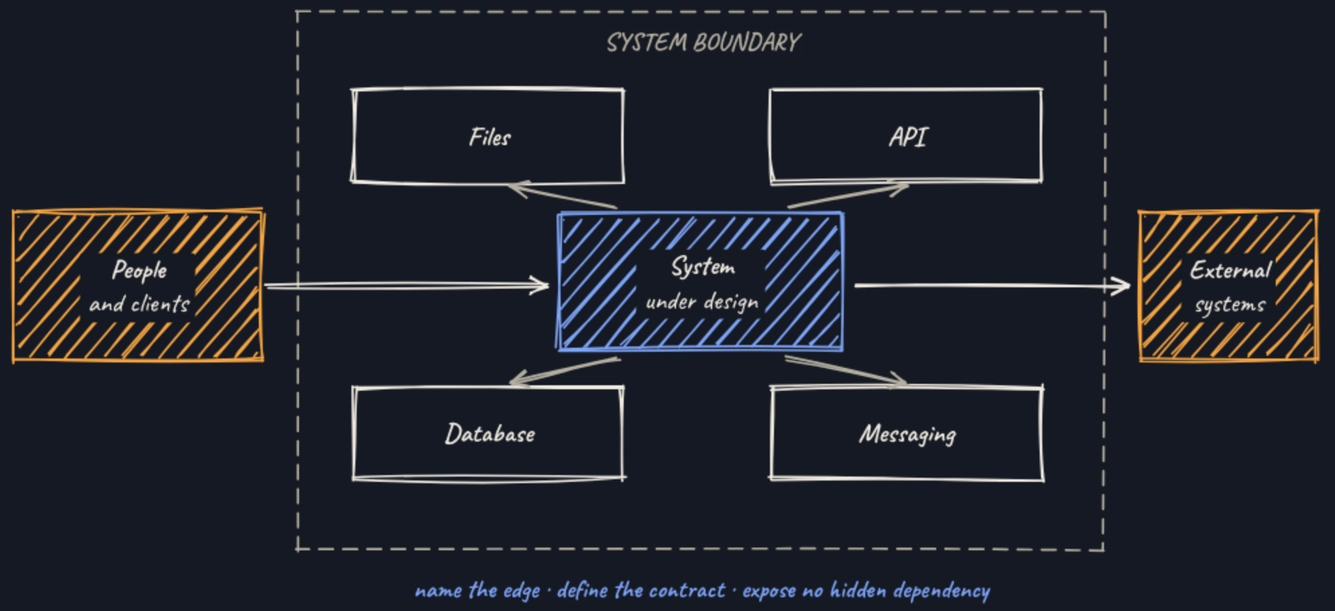
System boundaries

The system inside its environment

Integration 4 families

A boundary starts with a scenario

Files, DB, API, or Messaging



Understand every request path

From context to load balancer

- C4 System Context
- REST, RPC, GraphQL, AsyncAPI
- OSI, TCP/IP, DNS
- HTTP/1.1–HTTP/3 and WebSockets

Step 4

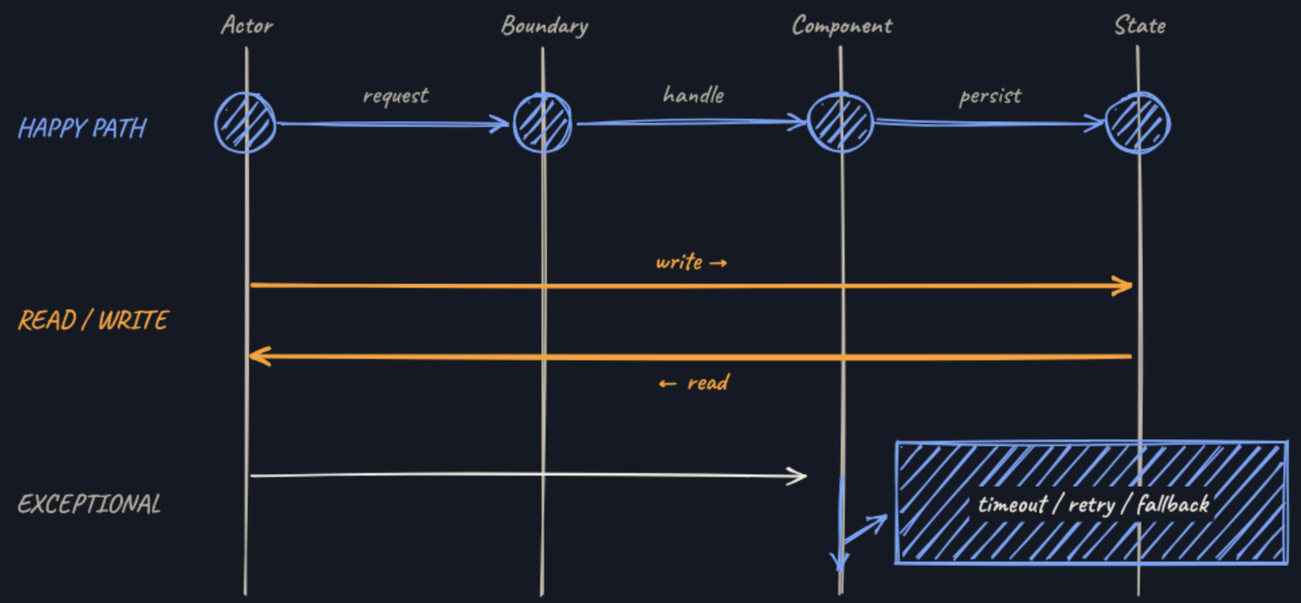
04

Flows

Components emerge from behavior

The happy path reveals exceptions

Then load and failing parts



Trace reads and writes

Validation Read / Write

Against the architecture characteristics

- Trace the read path
- Trace the write path
- Apply the leading NFRs
- Balance the read/write path

Choose the diagram for the question

Notation Conversation tool

What to study for flows

- C4 shows containers
- Sequence reveals interactions
- Activity captures branching
- IDEF0, DFD, and BPMN for flows

Step 5

05

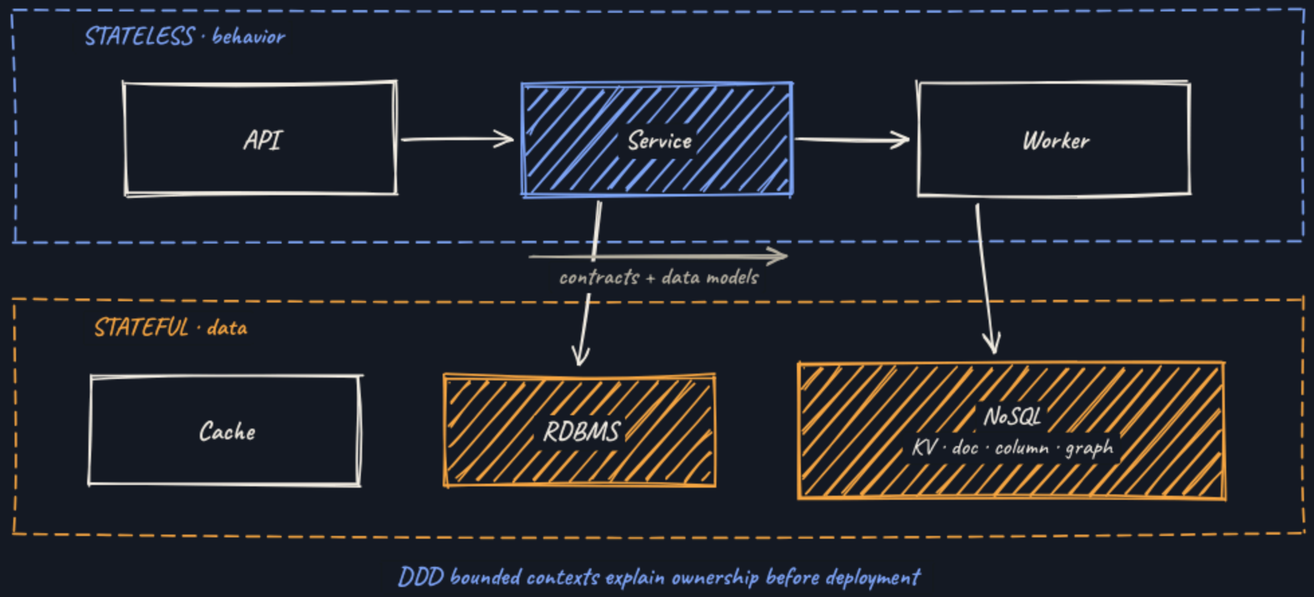
Conceptual design

Solution classes come first

Responsibility comes before product choice

Model No brands yet

Stateful, stateless, data, and contexts



Connect models to storage

Preparation | Data and domain

Data determines access

- RDBMS and NoSQL classes
- ER and Class Diagrams
- DDD and bounded contexts
- Twelve-Factor App

Step 6

06

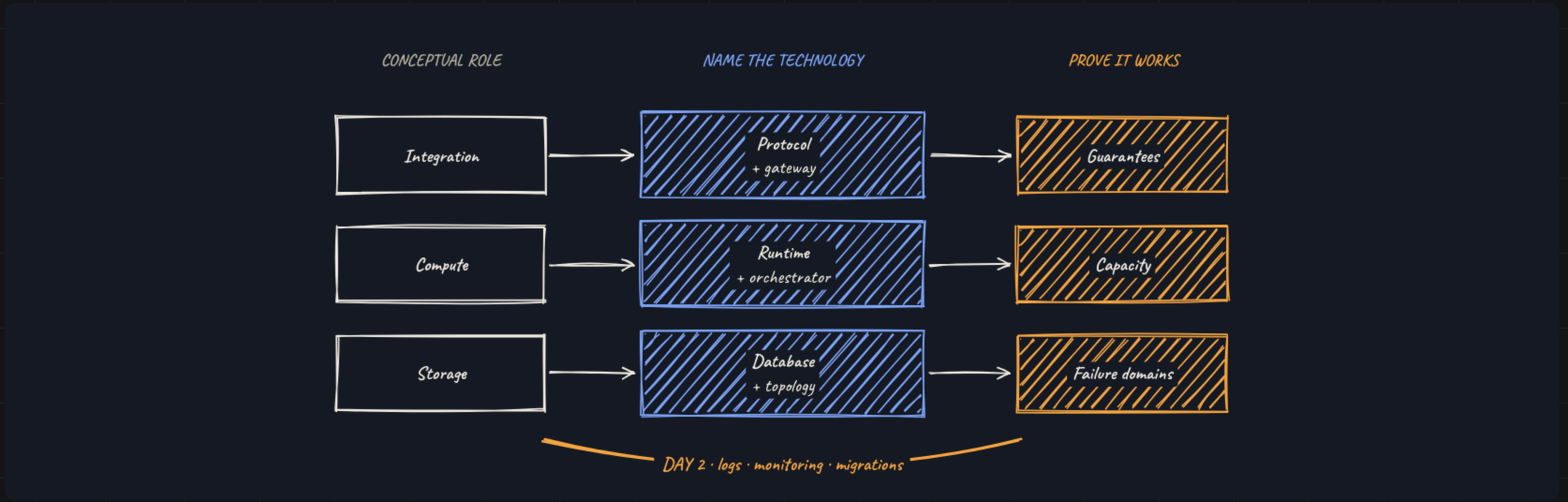
Real system design

Technology, guarantees, operations

Technology must uphold the promise

Guarantees, capacity, failure domains

Selection Concrete choices



Design continues after launch

Day 2 Operations

Operation and evolution

- Logs and monitoring
- Schema and data migrations
- Hands-on SRE tooling
- Google SRE books

Preparation · load

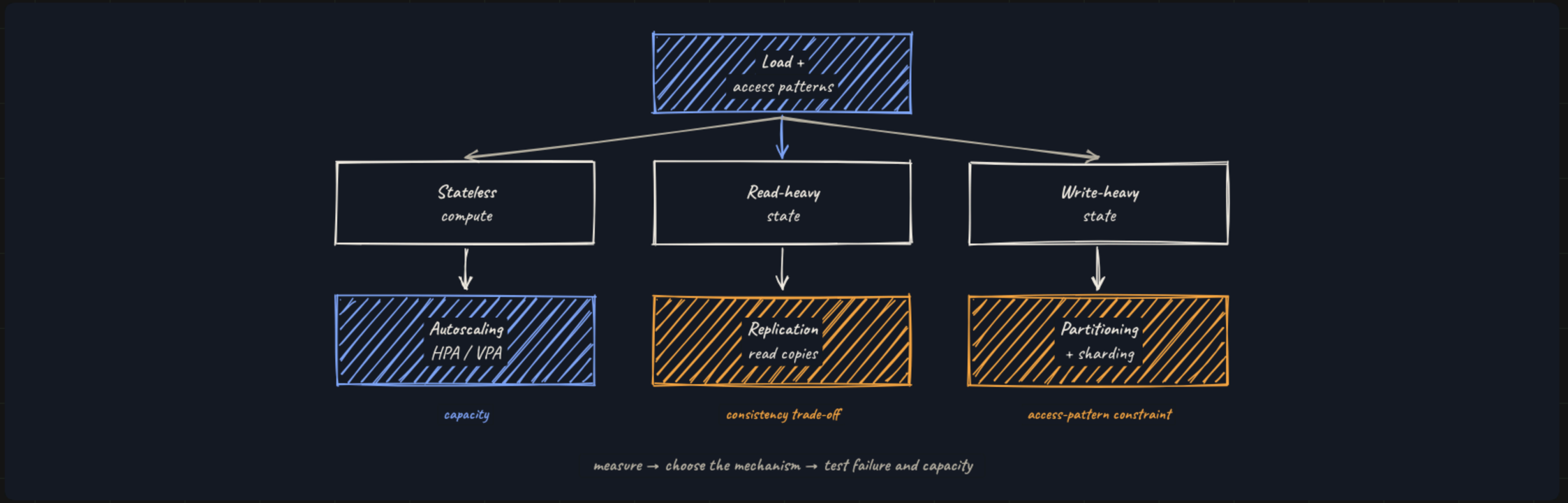
07

Under load

Different state, different levers

State determines the scaling mechanism

Orchestrators, replication, partitioning, sharding



Scale for real access patterns

Topology changes guarantees

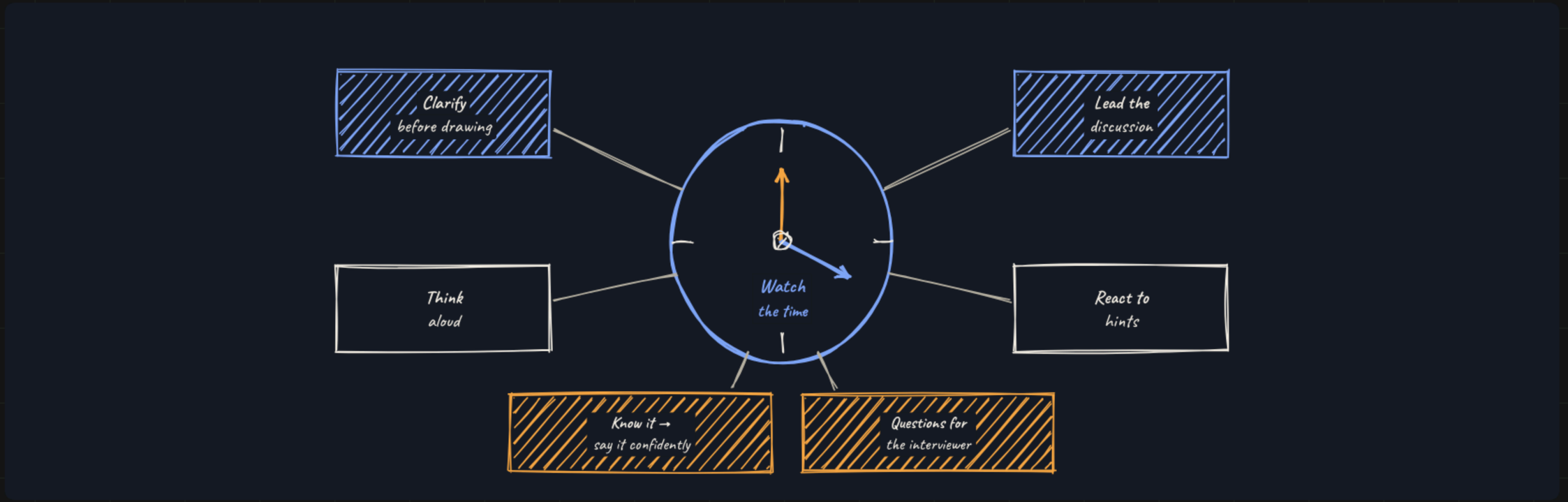
- Start with access patterns
- Name the consistency model
- Read DDIA and Database Internals
- Validate with load tests

Preparation Access patterns

Behavior `Lead the interview`

Lead the conversation, do not guess

Think aloud, respond to hints



Study along four paths

Resources 4 tracks

All links are in a Book Cube post

- Requirements and trade-offs
- Networks and integration
- Data and distribution
- Reliability and operations

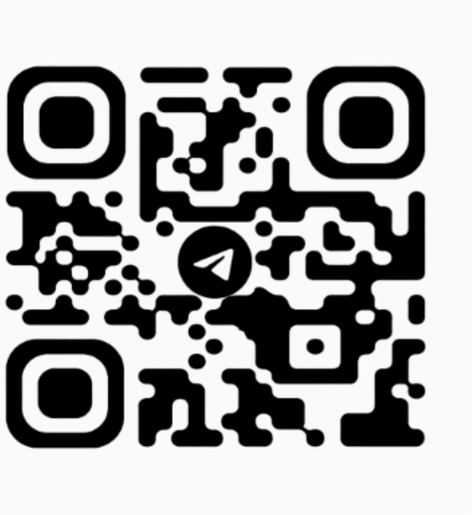
polomodov.tech

Meaning first, technology second

Clarify, trace flows, explain trade-offs

Alexander Polomodov

Author and speaker,
ArchDays 2022



@book_cube