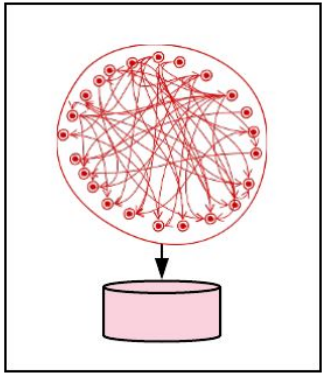
edX Technical Architecture

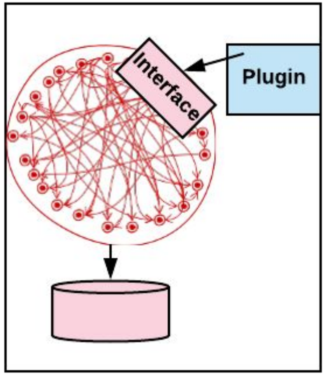
No Boundaries

Ball of Mud

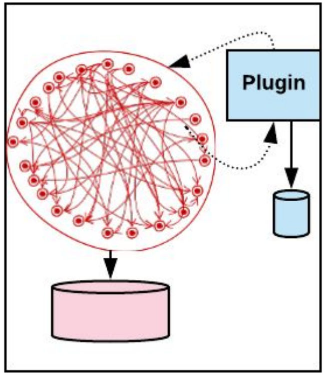


Logical Boundaries

Plugin with Interfaces

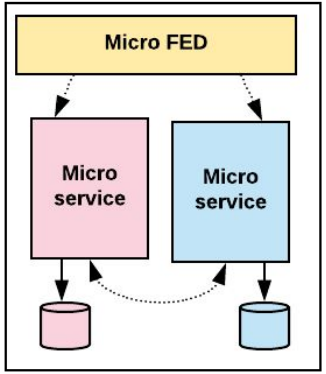


Plugin with Signals

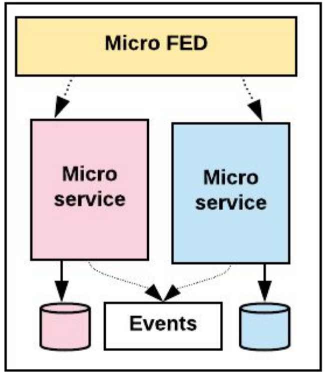


Physical Boundaries

DDD-based async Microservices

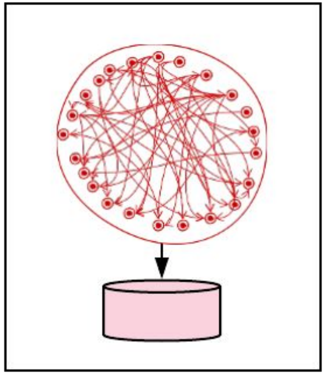


DDD-based eventing Microservices



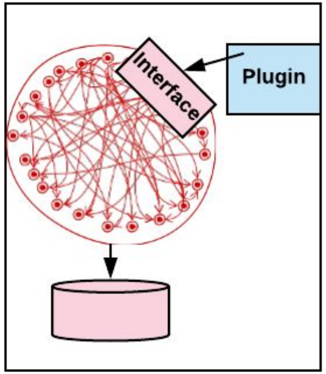
No Boundaries

Ball of Mud

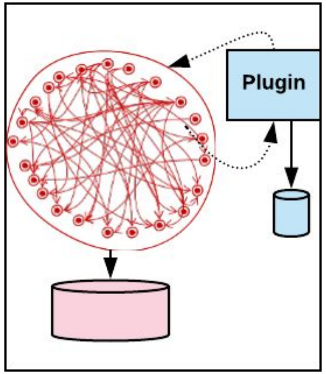


Logical Boundaries

Plugin with Interfaces

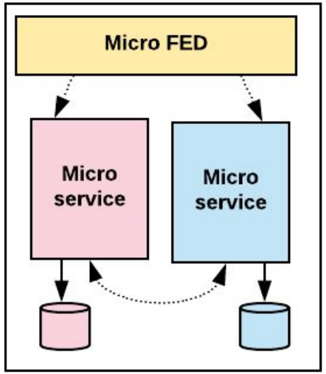


Plugin with Signals

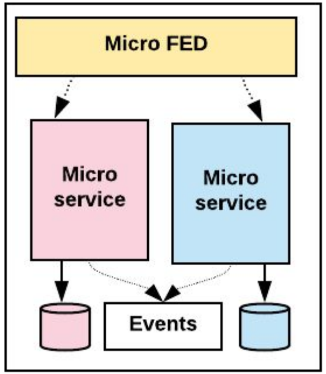


Physical Boundaries

DDD-based async Microservices



DDD-based eventing Microservices



Dependencies	Code	yes	yes	no	no	no
	Runtime	yes	yes	yes	no	no
	Deployment	yes	yes	yes	no	no
	Data	yes	maybe	no	no	no
	Interface	yes (python APIs)	yes (python interfaces)	yes (Django signals)	yes (REST, celery)	no (subscribers unknown)
	Testing	Mixed unit + integration	unit tests in plugin integration tests in main	consumer-based contract testing	consumer-based contract testing	consumer-based contract testing

No Boundaries

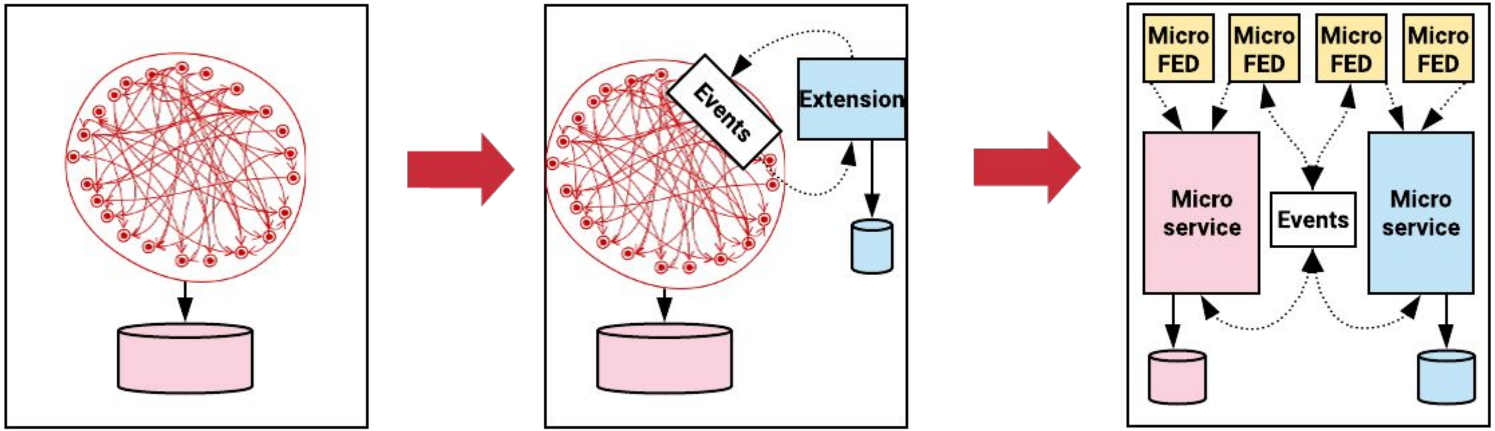
Monolith (Ball of Mud)

Logical Boundaries

Extending the Monolith

Physical Boundaries

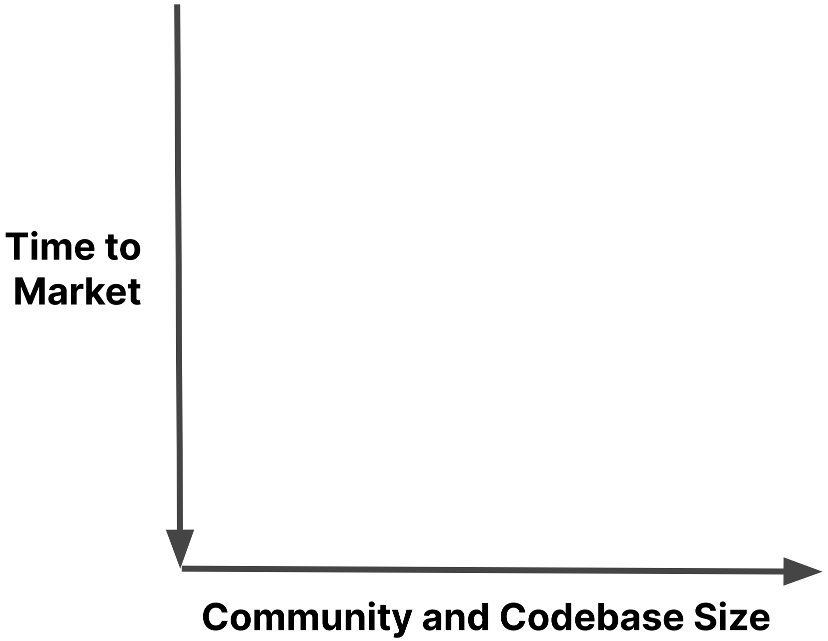
Small Decoupled Services

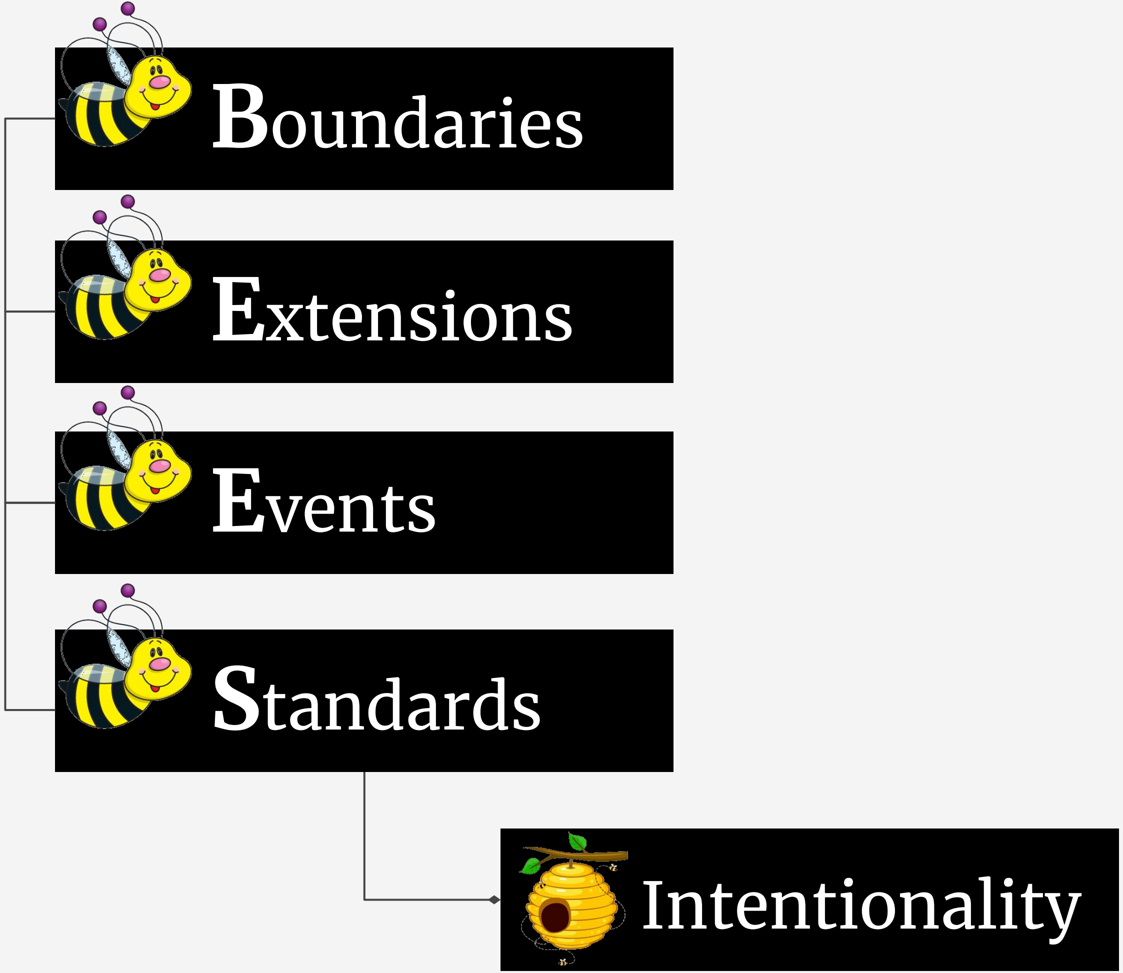


Cognitive Load	2.5 Million	5-100 Thousand	5-100 Thousand
Risk of Error	2.5 Million	2.5 Million	5-100 Thousand
Lead Time	2-4 hours	2-4 hours	10-30 mns
Reduced Autonomy	100+ (many engineers)	100+ (many engineers)	6-8 (owning team)

Extensible Core

Scaling Sustainably





Boundaries

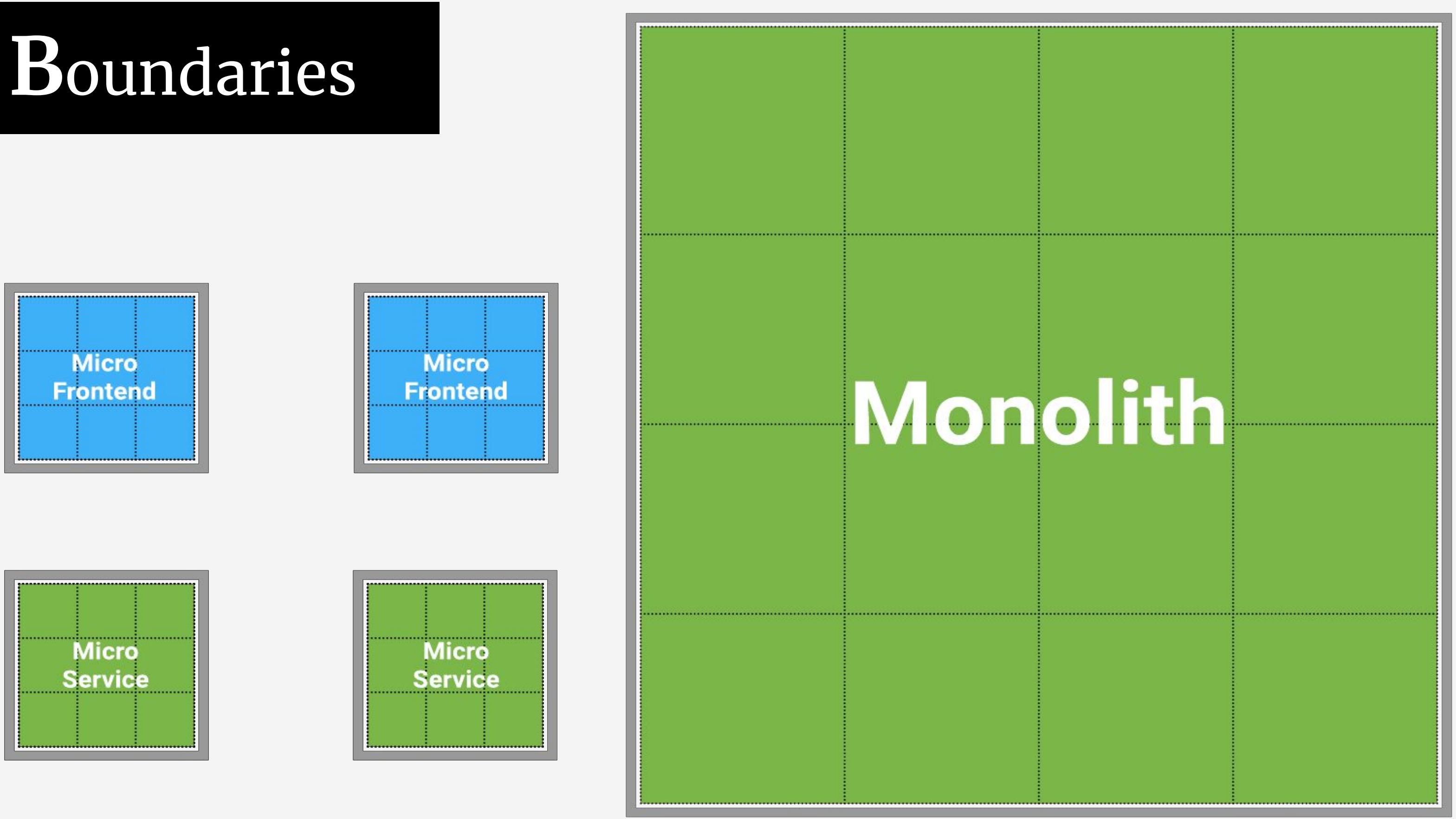
Extensions

Events

Standards

Intentionality

Boundaries



Micro Frontend

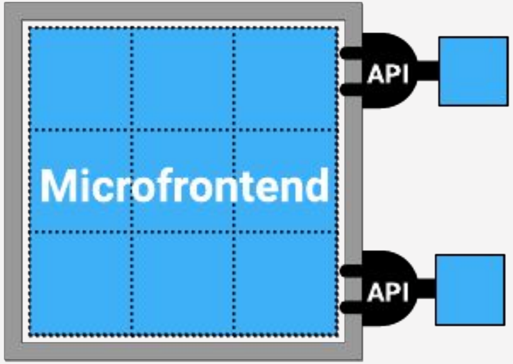
Micro Frontend

Micro Service

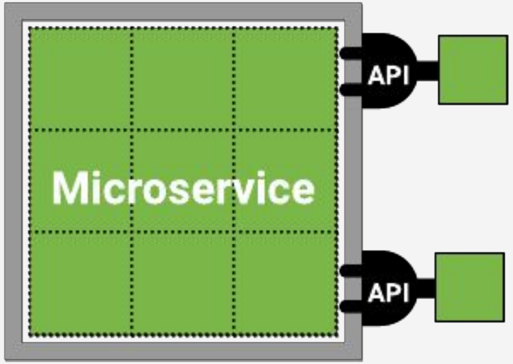
Micro Service

Monolith

Extensions

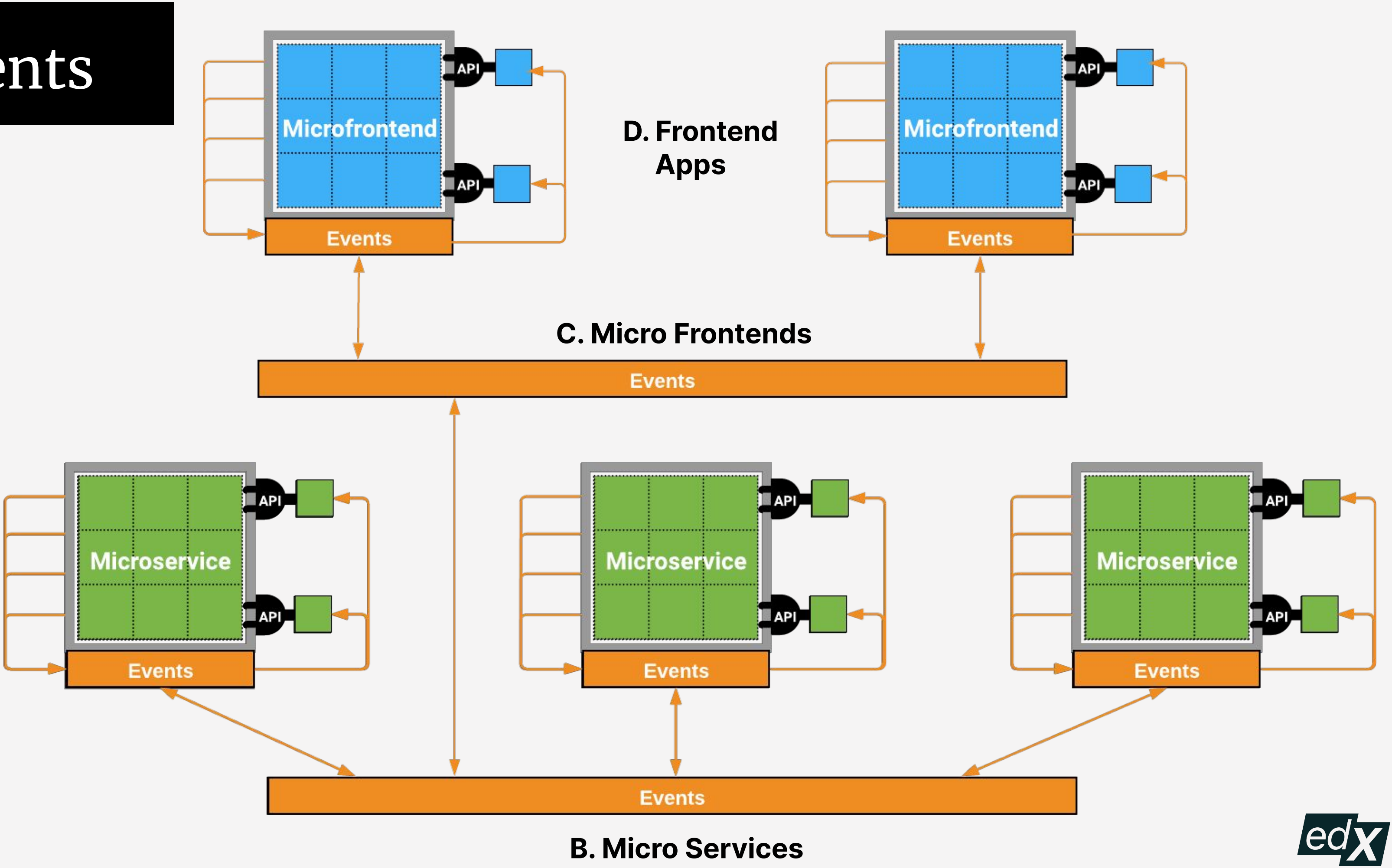


Frontend Plugin Framework
with Javascript APIs & Hooks



Backend Plugin Framework
with Python APIs & Hooks

Events


Microfrontend
API
API
Events
D. Frontend Apps
Microfrontend
API
API
Events
C. Micro Frontends
Events
A. Backend Apps
Microservice
API
API
Events
Microservice
API
API
Events
Microservice
API
API
Events
Events
B. Micro Services

Intentionality

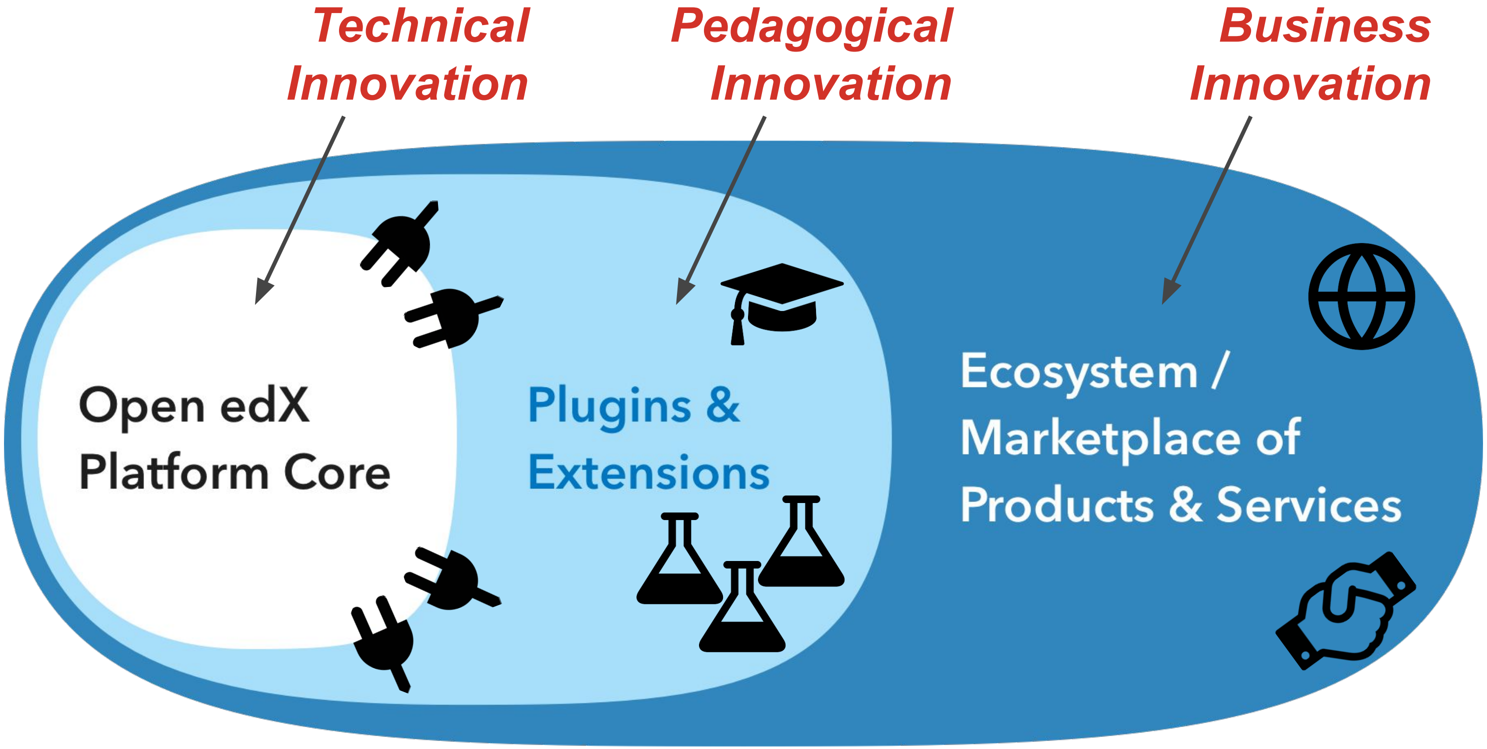
Intentional Architecture *beats* Emergent Design

Work to Simplify *beats* Slippery Complexity

Timeboxed Design *beats* `You Ain't Gonna Need It`

Decision Records *beat* Design Docs

Platform: Extensible Core



Platform: Extensions

Assessment Tools

xAPI / Caliper Events

Data Visualizations

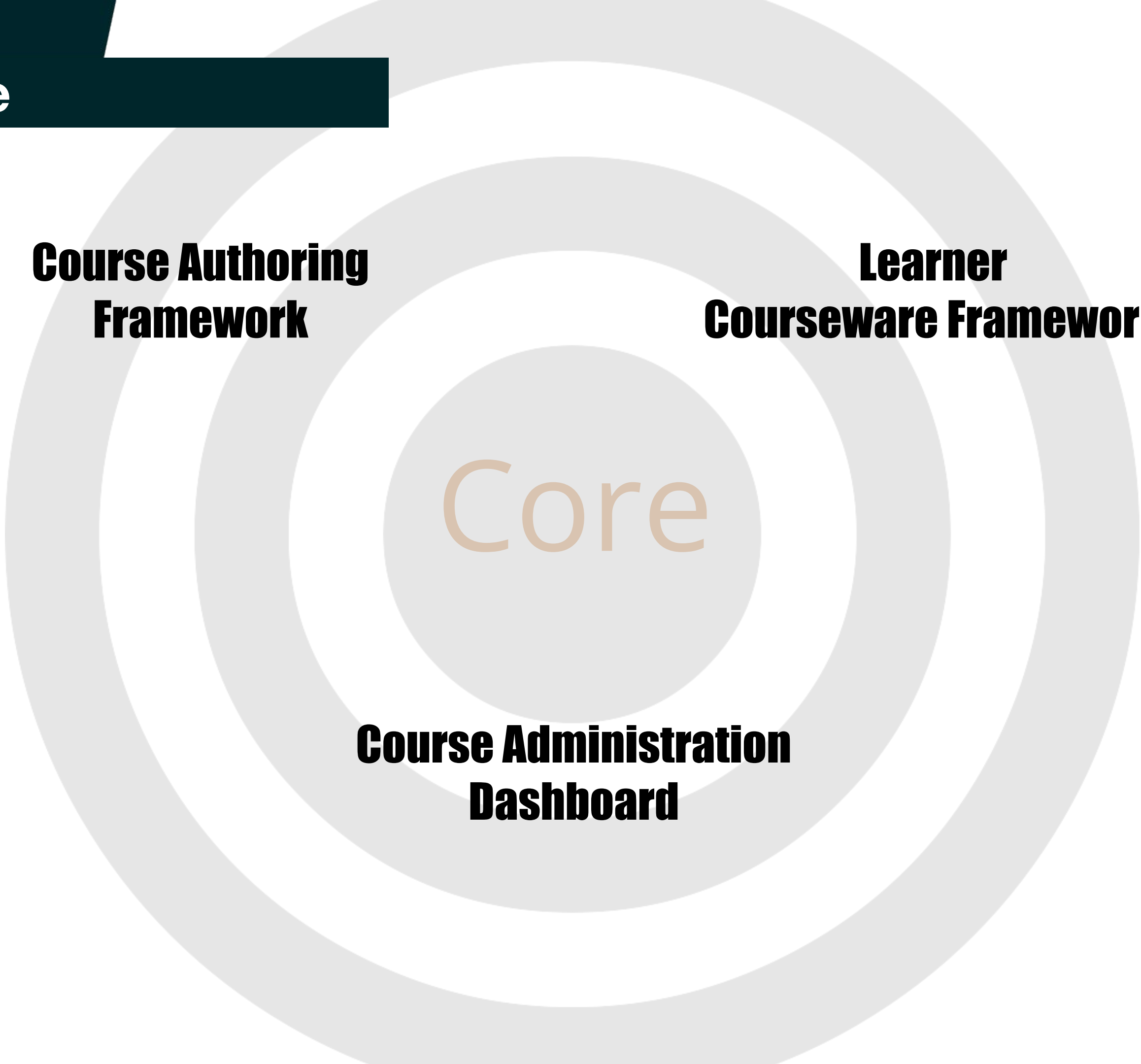
Plugins / Extensions

Credentialing Integrations

Adaptive Learning Experiences

Learning Support Tools

Platform: Core



Core

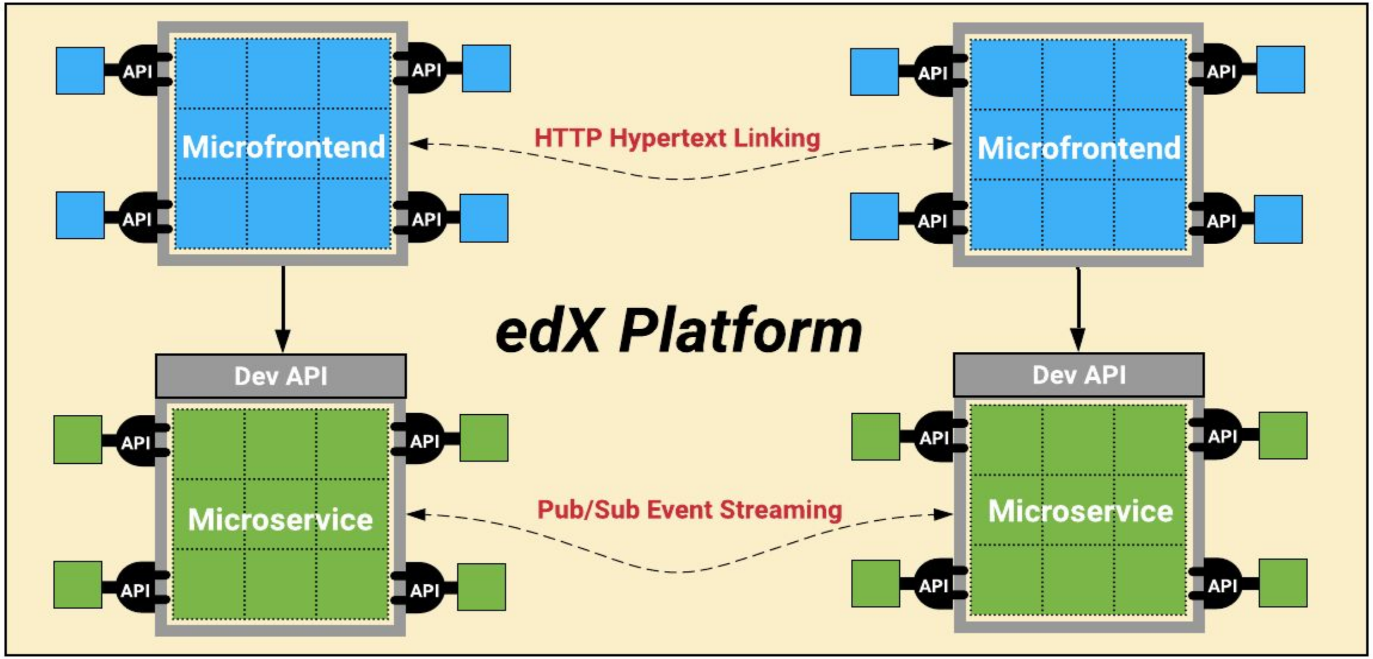
Course Authoring Framework

Learner Courseware Framework

Course Administration Dashboard

API Vision

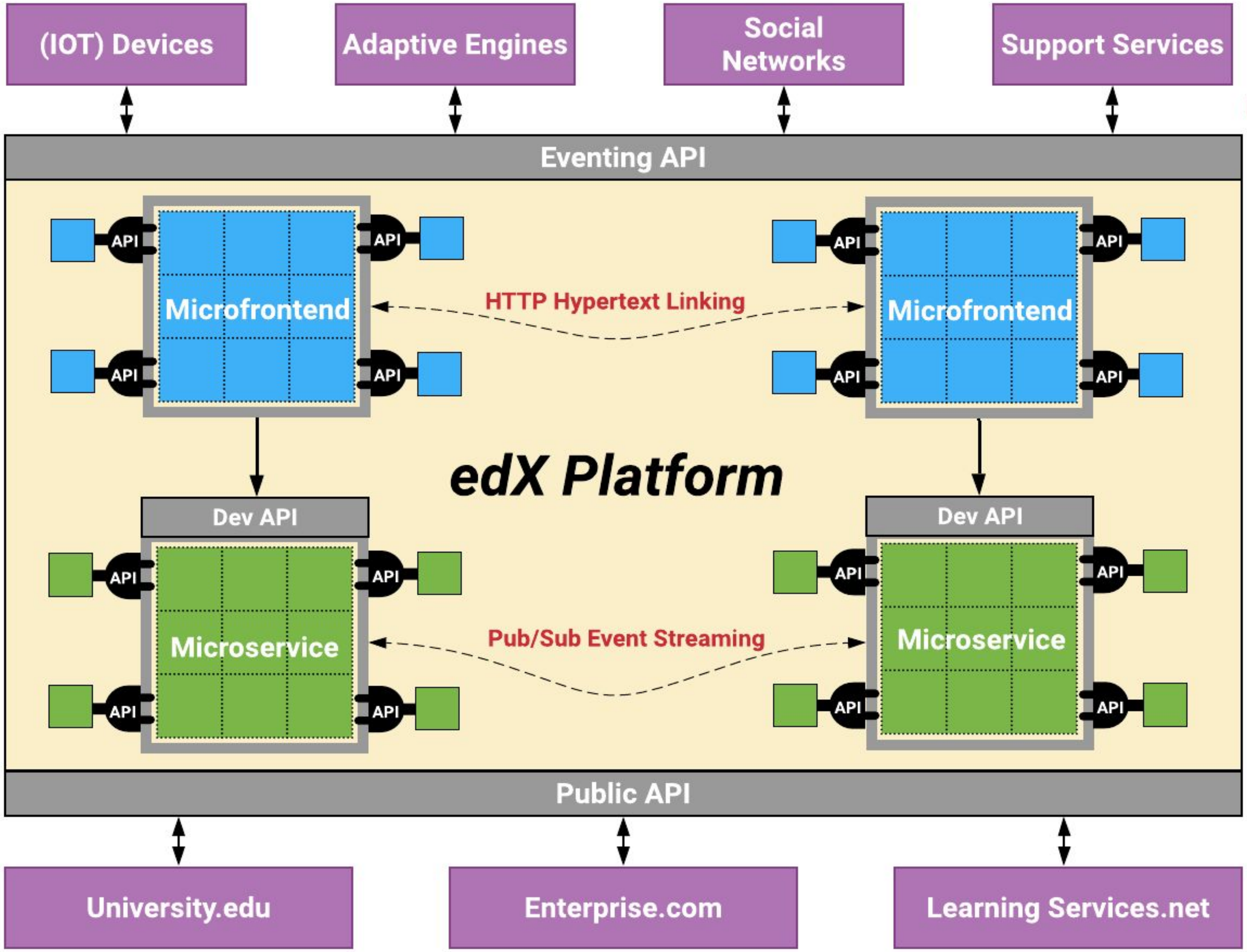
Platform: APIs



Frontend Plugins
Frontend Events
Supported JS APIs
Theming
  Branding
  Configuration
  Customization

Feature-driven REST / GraphQL APIs

Backend Plugins:
  Django App Plugins
  Django Web Fragments
  xBlocks, Block Transformers
Django Signals
Supported Python APIs



Real-time Events
xAPI, Caliper
Push Notifications or Pub/Sub

Frontend Plugins
Frontend Events
Supported JS APIs
Theming
Branding
Configuration
Customization

Feature-driven REST / GraphQL APIs

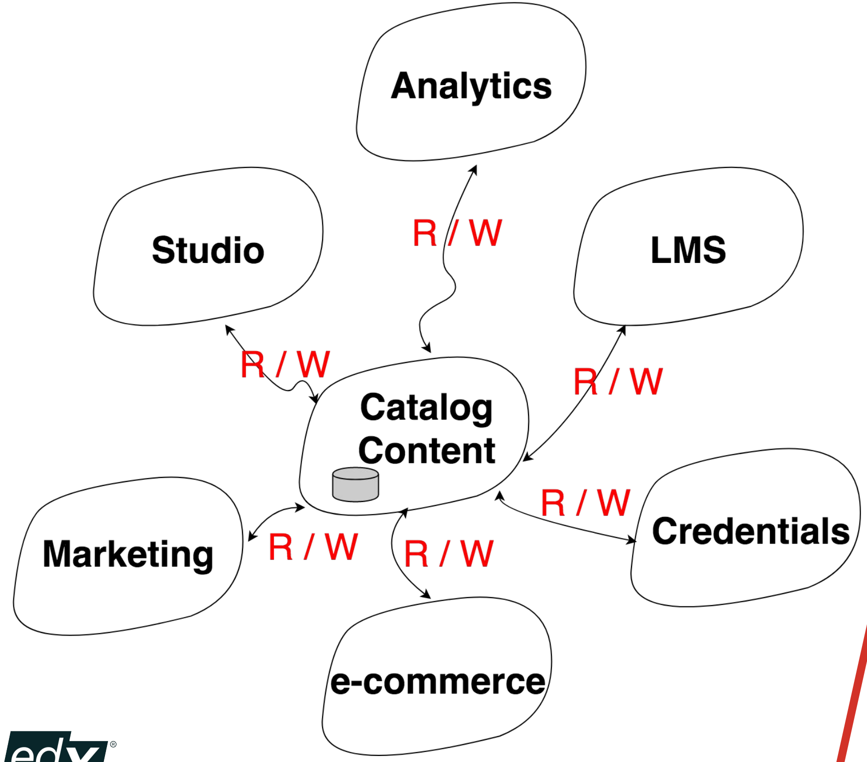
Backend Plugins:
Django App Plugins
Django Web Fragments
xBlocks, Block Transformers
Django Signals
Supported Python APIs

Service & Business-driven REST APIs
Content Integrations (OLX, LTI, Scorm)

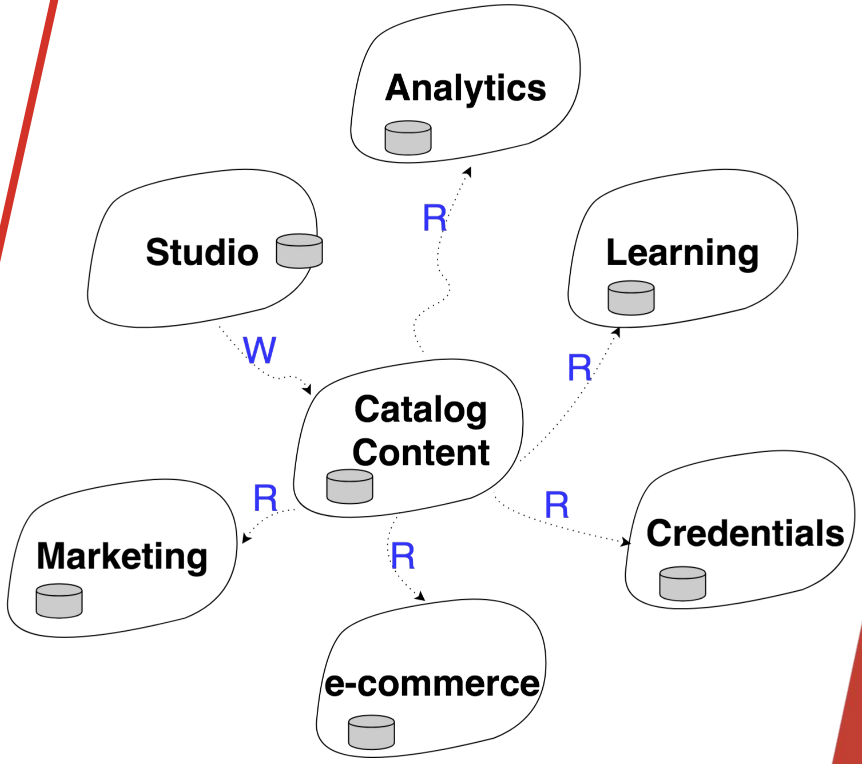
Architecture Manifesto & Principles

Distributed Data

Option A

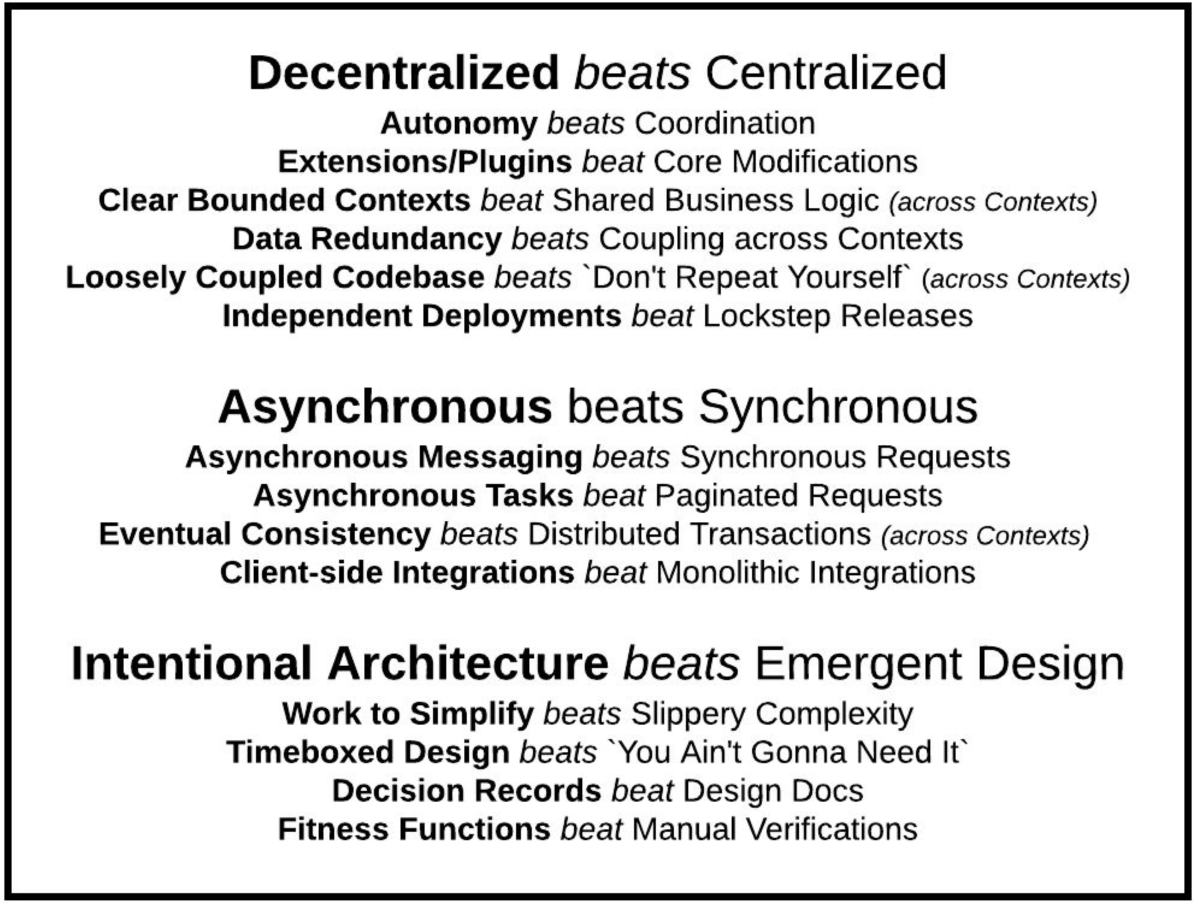


Option B



edX Architecture Manifesto

(link)



Decentralized *beats* Centralized

Autonomy *beats* Coordination
Extensions/Plugins *beat* Core Modifications
Clear Bounded Contexts *beat* Shared Business Logic *(across Contexts)*
Data Redundancy *beats* Coupling across Contexts
Loosely Coupled Codebase *beats* `Don't Repeat Yourself` *(across Contexts)*
Independent Deployments *beat* Lockstep Releases

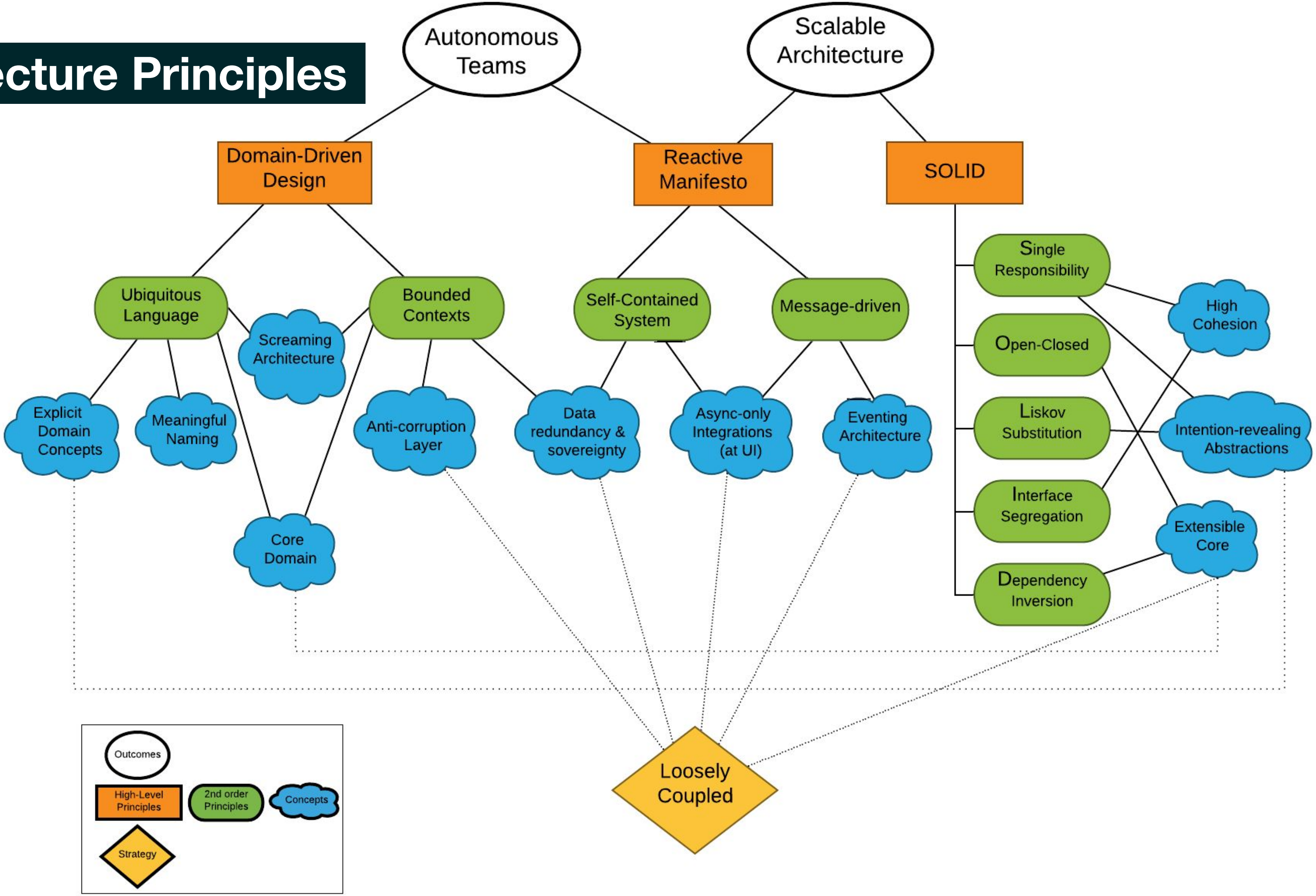
Asynchronous beats Synchronous

Asynchronous Messaging *beats* Synchronous Requests
Asynchronous Tasks *beat* Paginated Requests
Eventual Consistency *beats* Distributed Transactions *(across Contexts)*
Client-side Integrations *beat* Monolithic Integrations

Intentional Architecture *beats* Emergent Design

Work to Simplify *beats* Slippery Complexity
Timeboxed Design *beats* `You Ain't Gonna Need It`
Decision Records *beat* Design Docs
Fitness Functions *beat* Manual Verifications

Architecture Principles



Thank you!

Any questions?