Accionlabs

Accion INNOVATION SUMMIT 2023

02-05 March 2023,
Sofitel Dubai
The Palm Jumeirah
Dubai

Accion INNOVATION SUMMIT 2023

Accionlabs

Complexity Management through Architecture Patterns

Saurabh Periwal

Chief Architect

As part of my job, I work closely with teams to

1. Help achieve technology goals
2. Roadmap technology strategy
3. Establish market position
4. Oversee important technology initiatives.

I often address following concerns

1. Each business is different, it involves unique complexity, because …
2. How complex is our architecture?
3. We need a microservice architecture?

Problem Statement:

Matching architecture complexity to business complexity

Background

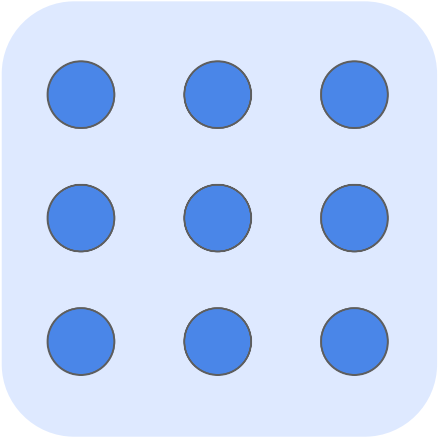
- **Business Complexity**
  - Difficulty involved in managing and operating business.
  - Factors
    - Product and services diversity
    - Regulatory environment
    - Scope of operations

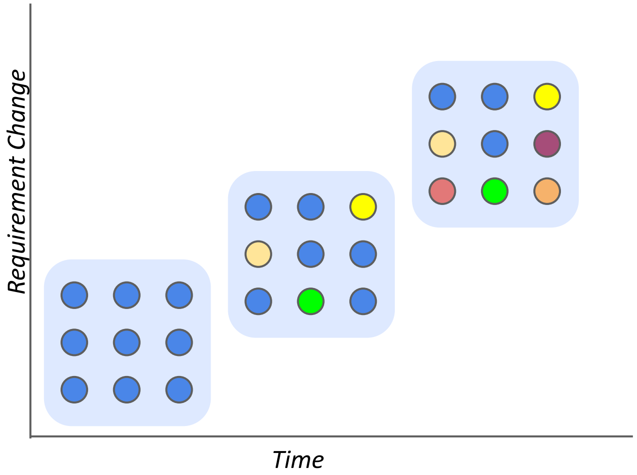
Background

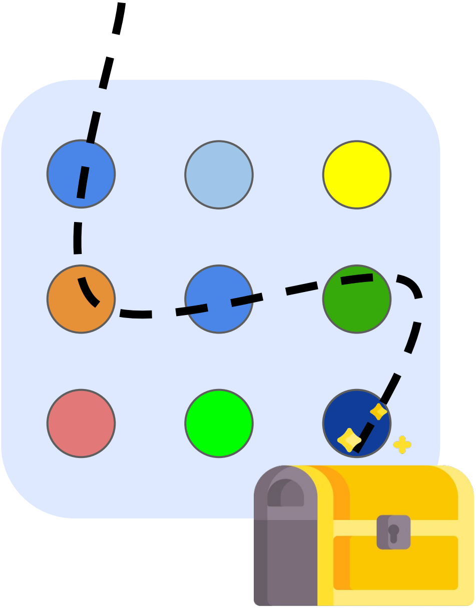
Accionlabs

- **Software Architecture Complexity**
  - Difficulty involved in designing, implementing and maintaining.
  - Factors
    - components interact and communicate with one another.
    - size, scope, requirements of the software system,
    - technologies
    - programming languages used to build it.

Types of Complexity







Substance Complexity
Number of different components or processes required to build a product

Dynamic Complexity
Rate of change in components or process in product / service

Psychological Complexity
Human interaction and behaviour with in a system

Business Complexity

Substance complexity, dynamic complexity with Business complexity

e-Commerce Portal

Website to buy and sell products

1. New Product Category and attributes.
2. Search & Browse User flows
3. Product configurations
4. New Payment Methods
5. New Shipping Methods
6. New Geographies
7. New Tax Implications on goods
8. New Product Vendors

Product Catalog
Inventory Management
Pricing Management
Payment Processing
Shipping & Logistic
Returns Management
Marketing and Personalization
User Profile Management

processing, payment gateways, shipping logistics, marketing.	optimize for usability, personalization, recommendations
Moderate Substance Complexity	High Dynamic Complexity

Internet Banking

A digital platform that allo activities and transactions

Regulatory compliance, se management, and the nee

1. Fund Transfers
2. Bill Payments
3. Loan Management
4. Investment Management
5. Wealth Management
6. Personal Finance
7. Trading Account
8. TAX Payments
9. Customer Service
10. Security Management

Account Information
KYC
Anti Money Laundering

1. Security Features
2. Bill Payments
3. Integrations with third party.

KYC, AML	transactions, deposits, withdrawals, transfers, investments
High Substance Complexity	Low Dynamic Complexity

Summary

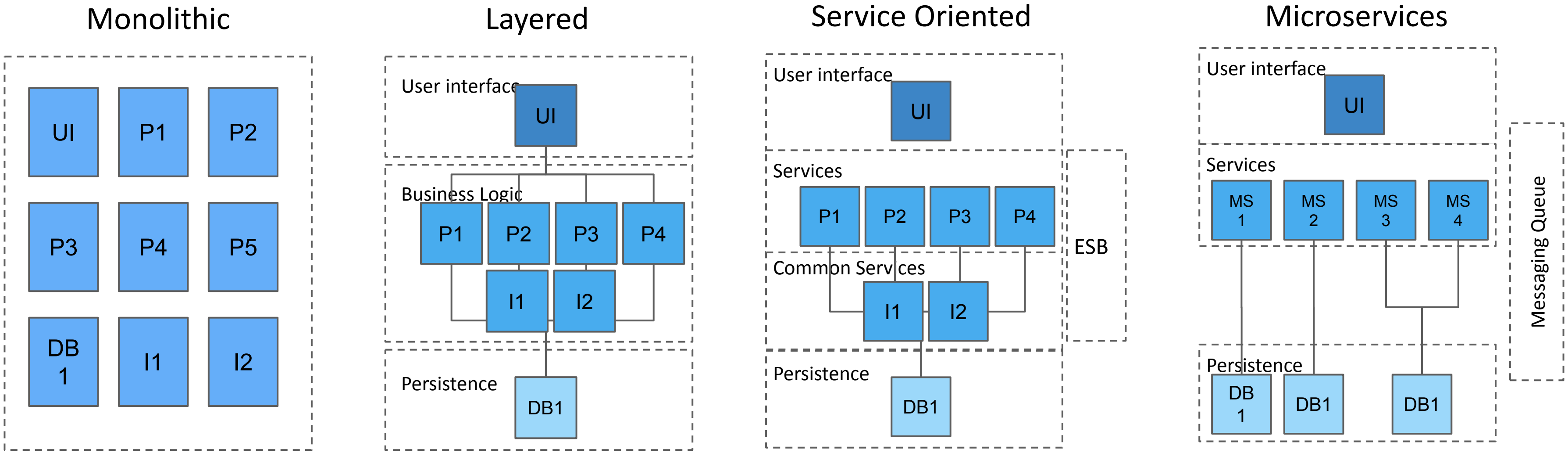
The complexity of the business often drives the complexity of the underlying architecture, as the architecture must be designed to accommodate the various components and interactions of the business.

Architecture Patterns

Complexity Assessment for Architecture Patterns

	Complexity of Architecture	Effort Required to Handle Complexity of Business Requirements
Substance Complexity	Number of independently deployable components	Effort Required to Add Features
Dynamic Complexity	Flexibility of components	Effort Required to Change Features

Conventional Architectural Models



Monolithic

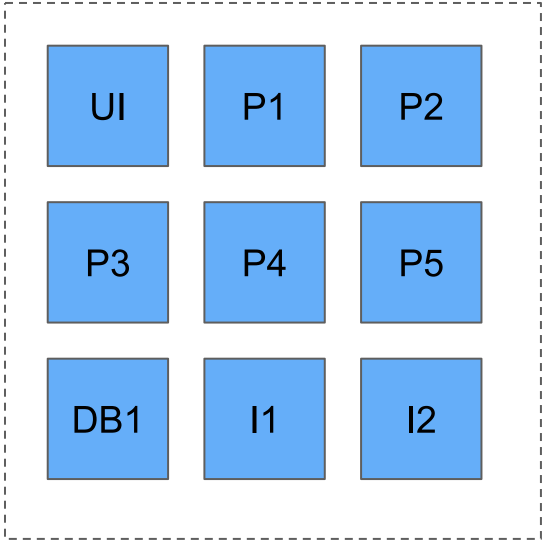
Layered

Service Oriented

Microservices

Conventional Architecture Models are implemented based on the required level of business complexity

Architectural Pattern - Monolithic



Monolithic Architecture	Complexity of Architecture	Effort to Handle Complexity of Business Requirements
Substance Complexity	Low (Typically single deployable unit)	High
Dynamic Complexity	Low (Any change requires extensive testing)	High

Substance Complexity 🟩

All the components are packaged together into a single, tightly-coupled application.

Dynamic Complexity 🟩

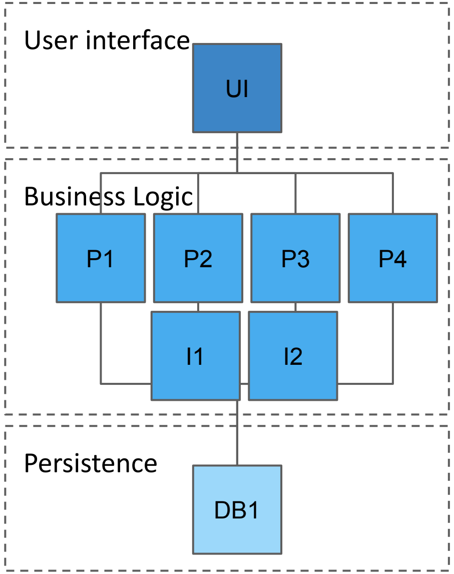
Difficult to manage and scale the interactions and dependencies between the components. Any change require extensive testing

Suitable for Business

low rate of change are those that operate in stable and predictable environments, and can typically rely on more traditional and established architectures to meet their needs

- Traditional Manufacturing Companies
- Utilities
- Government Agencies

Architectural Pattern - Layered



Substance Complexity 🟧

The architecture separates different components or layers into distinct and modular units.

Dynamic Complexity 🟩

Layers in a system communicate via defined interfaces. Changing requirements or functionality of each layer can alter the system's behavior, affecting the entire system

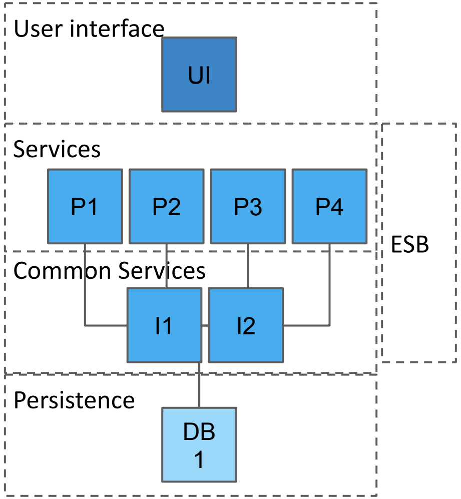
Layered Architecture	Complexity of Architecture	Effort to Handle Complexity of Business Requirements
Substance Complexity	Moderate	High
Dynamic Complexity	Low	High

Suitable for Business

Overall, businesses with low substance complexity and moderate dynamic complexity are those that require architectures that are responsive, scalable, and flexible enough to handle periodic updates or changes to the application logic, but do not require rapid changes to the application behavior.

- Low Traffic e-Commerce
- Online booking
- CRM

Architectural Pattern - Service Oriented



SOA	Complexity of Architecture	Effort to Handle Complexity of Business Requirements
Substance Complexity	Moderate (3 or more deployable units)	High
Dynamic Complexity	Moderate (Any change requires extensive testing)	Moderate

Substance Complexity

while it simplifies the overall architecture by breaking it down into smaller components, it still requires expertise and management to ensure that the system functions correctly and meets the desired requirements.

Dynamic Complexity

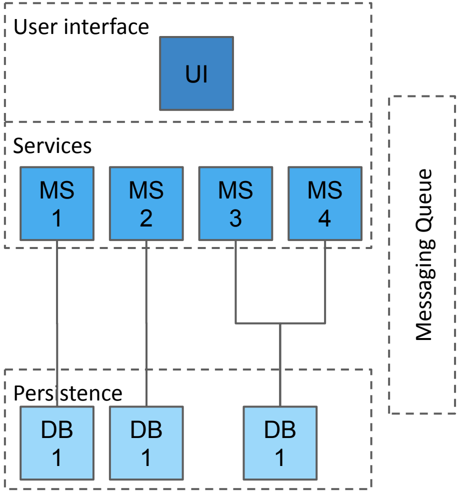
it allows for flexibility and adaptability, it still requires careful management and ongoing maintenance to ensure that the system functions correctly and meets the desired requirements.

Suitable for Business

Typically businesses that involve relatively straightforward products or services with some degree of regulation or management complexity.

- Education
- Real Estate

Architectural Pattern - Microservices



Microservice	Complexity of Architecture	Effort to Handle Complexity of Business Requirements
Substance Complexity	High	Low
Dynamic Complexity	Low	Low

Substance Complexity 🟥

Microservices break down a system into small, independent services responsible for specific functions. Services communicate through APIs, providing scalability

Dynamic Complexity 🟩

Each service in a microservices architecture is responsible for a specific function. It is predictable how changes in one service may affect the behavior of the entire system

<u>Businesses with high Substance and Dynamic Complexity.</u>

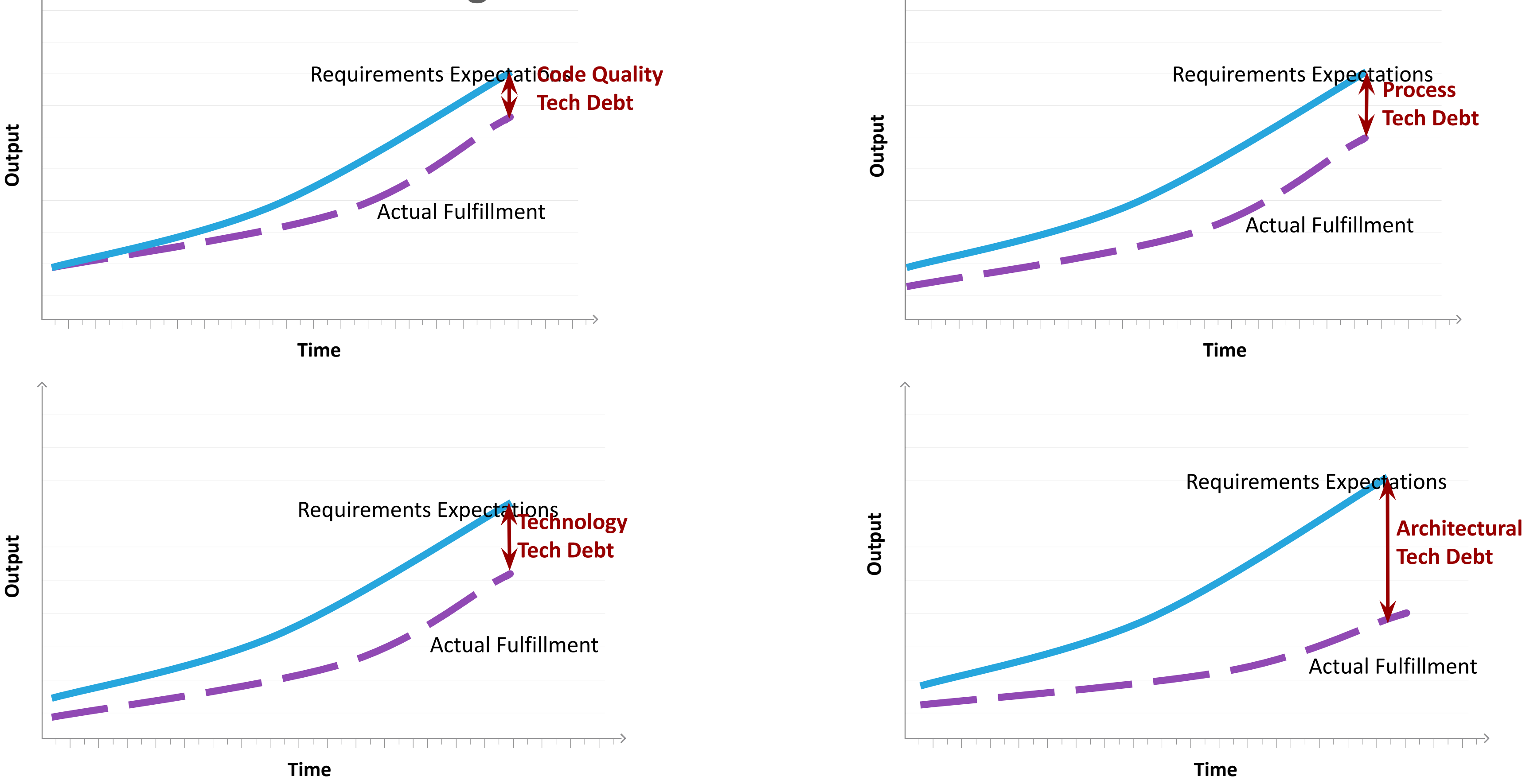
Businesses that deal with large-scale, <u>complex systems and processes that are subject to constant rapid change</u>.

- Banking
- High Traffic e-commerce
- Telecommunications.

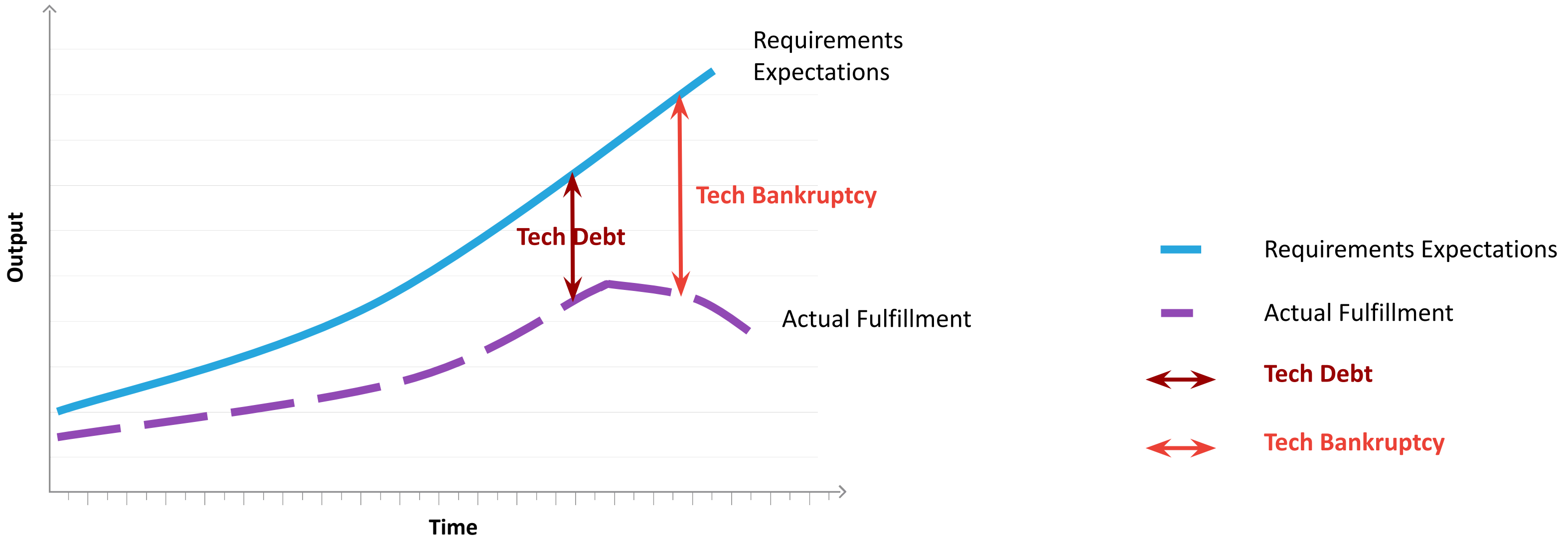
Mapping of Architecture Complexity and Business Complexity

Architectural Pattern	Substance Complexity		Dynamic Complexity		Suitable Business Conditions
	Architectural Complexity	Effort / Business Complexity	Architectural Complexity	Effort / Business Complexity	
Monolithic	Low	High	Low	High	Stable, predictable e.g. manufacturing
Layered	Moderate	High	Low	High	Periodic update E.g. niche ecom, crm,
Service Oriented	Moderate	High	Moderate	Moderate	Some degree of regulation and management e.g education, realestate
Microservices	High	Low	High	Low	Constant rapid change Large Scale E.g. banking, high-traffic ecom

Architecture Debt is the Highest Contributor to Technical Debt



Architectural Technical Debt can cause Technical Bankruptcy



Technical Bankruptcy occurs when system starts delivering negative Output as the time and efforts are consumed in fixing the system complexity (entropy / tech debt)

Architectural misalignment is the primary cause of Technical Bankruptcy

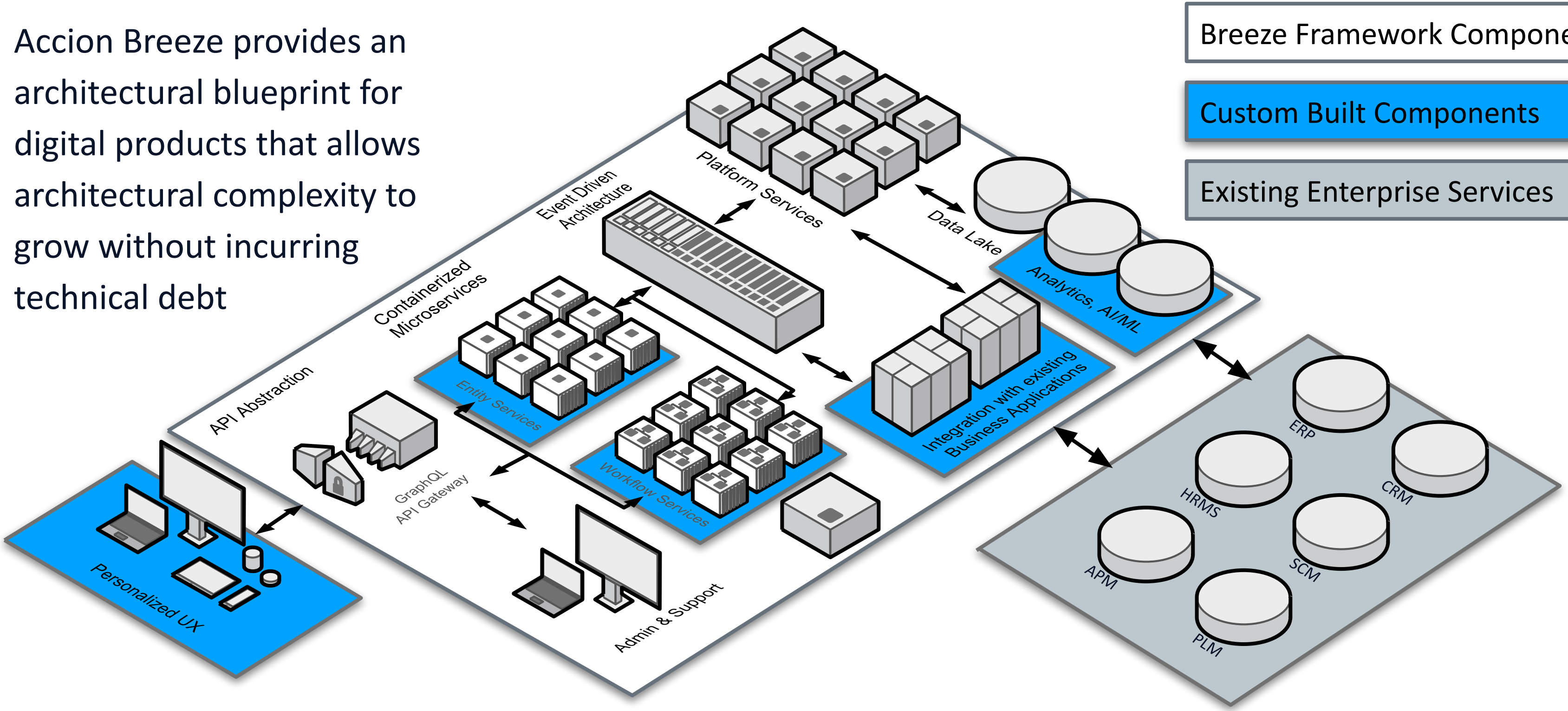
Accion Breeze - Digital Product Architecture Blueprint

Accion Breeze provides an architectural blueprint for digital products that allows architectural complexity to grow without incurring technical debt

Breeze Framework Components

Custom Built Components

Existing Enterprise Services



Accion Breeze - Minimum Viable Architecture Complexity

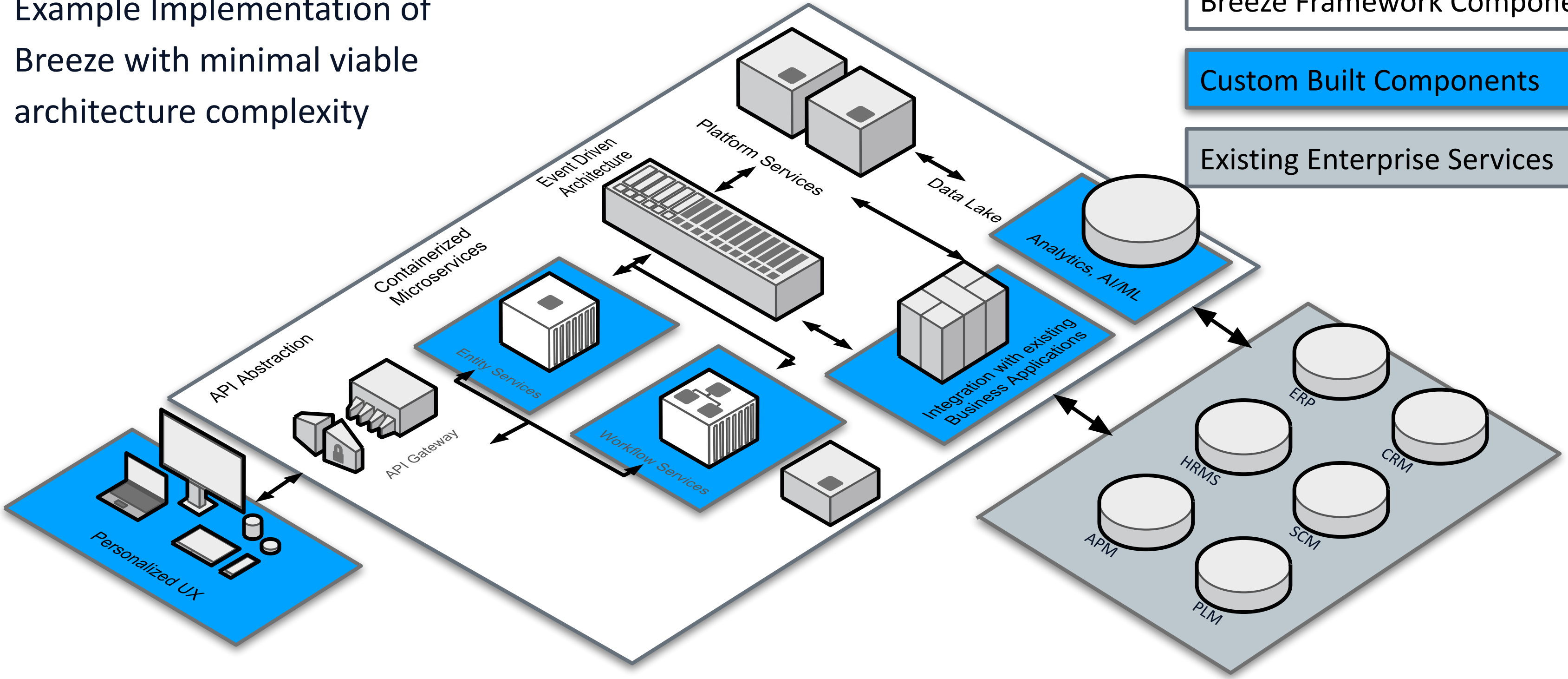
Accionlabs

Example Implementation of Breeze with minimal viable architecture complexity

Breeze Framework Components

Custom Built Components

Existing Enterprise Services



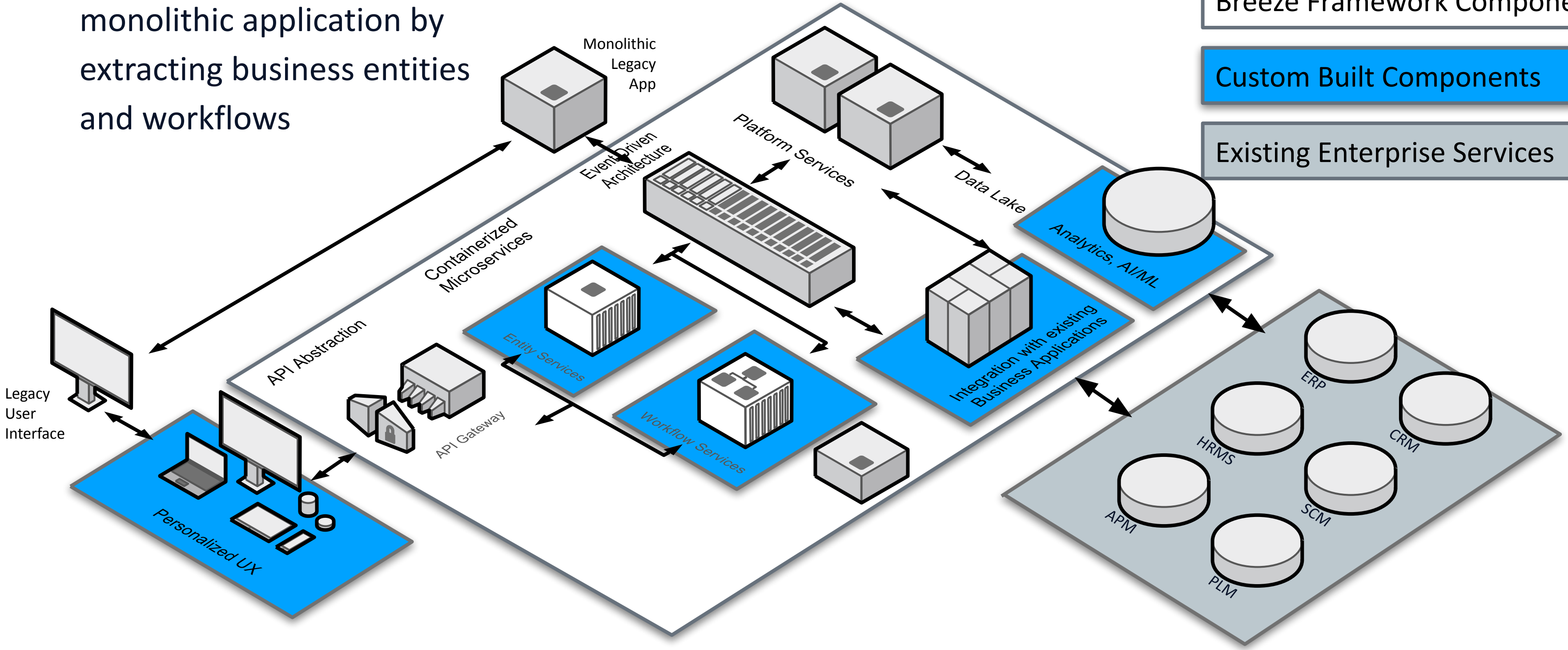
Accion Breeze - Re-engineering Path of Least Complexity

Example for re-engineering a monolithic application by extracting business entities and workflows

Breeze Framework Components

Custom Built Components

Existing Enterprise Services



Thank you