Product Portfolio Rationalization

Methodology & Case Studies

Accionlabs

Product Portfolio Analysis

Background

What are the **motivations** that lead to creation of a Product Portfolio?

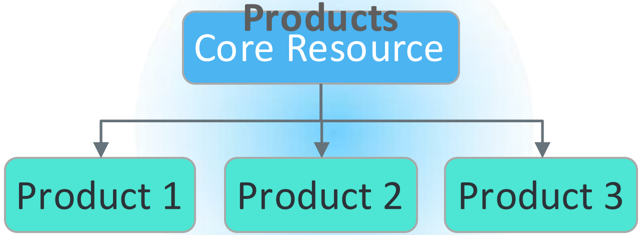
A. **To diversify product offerings**
B. **Segment across different geographies / demographics etc.**
C. **Create additional revenue streams using existing offerings / resources**
D. **Adjunct products for innovation**

Methodologies to create a product portfolio?

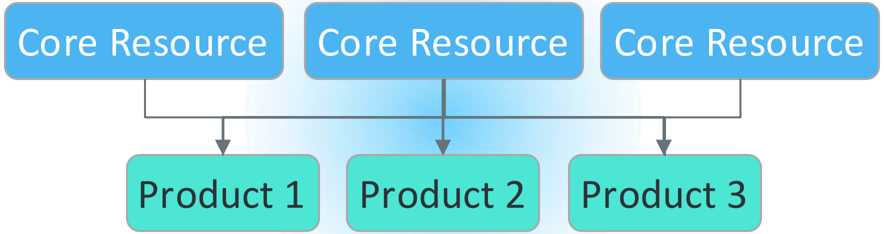
A. **Inorganic Evolution -** Mergers and Acquisitions
B. **Organic Evolution -** Self Developed Solutions

Structural Patterns in Product Portfolio

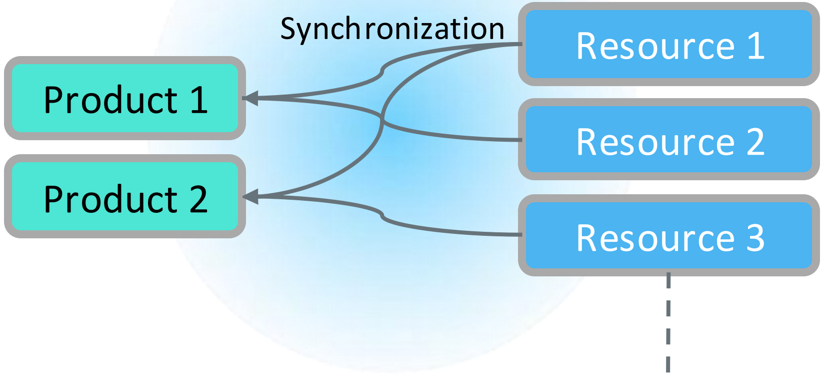
Core Resource Common Between Products



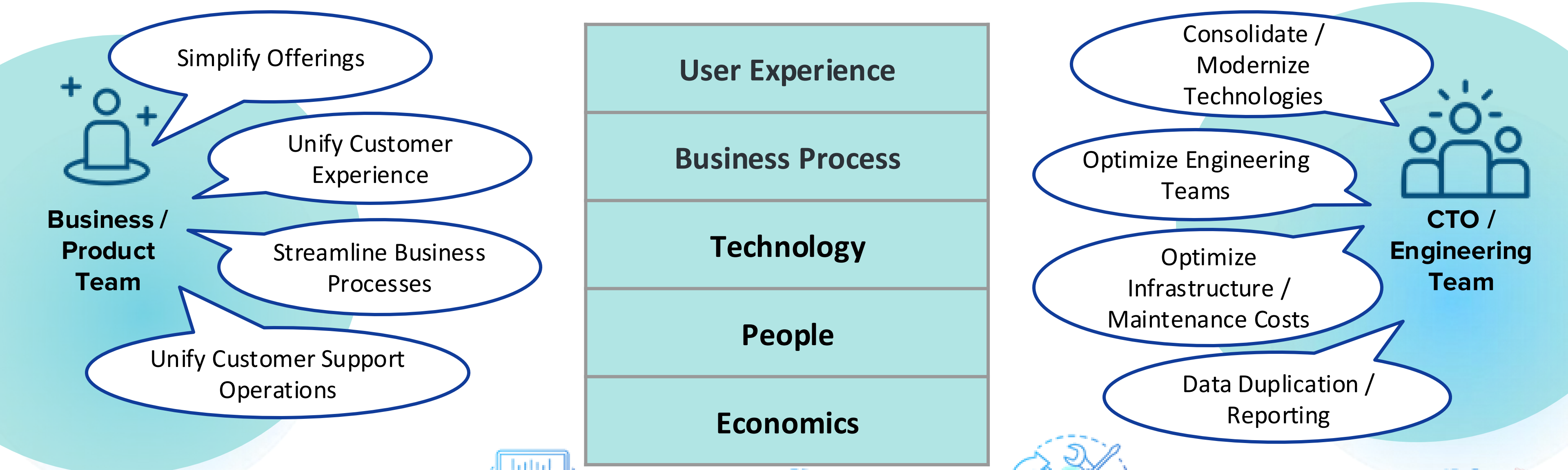
Multiple Resources Common Between Products



Synchronization Between Resources



Business & Technology Objectives



Business / Product Team

- Simplify Offerings
- Unify Customer Experience
- Streamline Business Processes
- Unify Customer Support Operations

User Experience

Business Process

Technology

People

Economics

CTO / Engineering Team

- Consolidate / Modernize Technologies
- Optimize Engineering Teams
- Optimize Infrastructure / Maintenance Costs
- Data Duplication / Reporting

Key Pain Points Summarized

Category	Root Cause
I can’t run an international influencer marketing campaign that spans multiple geographies from a single portal.	Functional Overlap
I have a vintage tractor that I need to auction, but have to manage and run the auction on multiple websites.	
As a Product Manager, everytime I have to modify a feature, I have to modify it in all the products.	
Our Infrastructure costs are too high considering all the applications that we have to host.	Architecture Overlap
Everytime I switch from one product to another I have to login again.	
Everytime we find a new data source to integrate into our products, all of our products have to undergo changes for that integration.	

Accion's Rationalization Methodologies

Rationalization Tenets

Functional Overlap

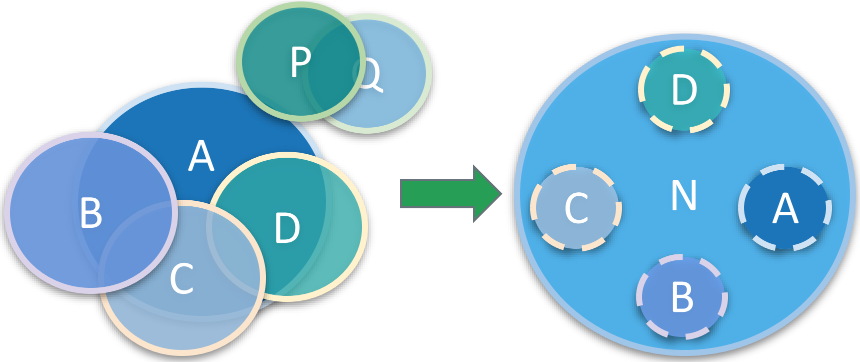
Consolidate → Migrate → Sunset

Architecture Overlap

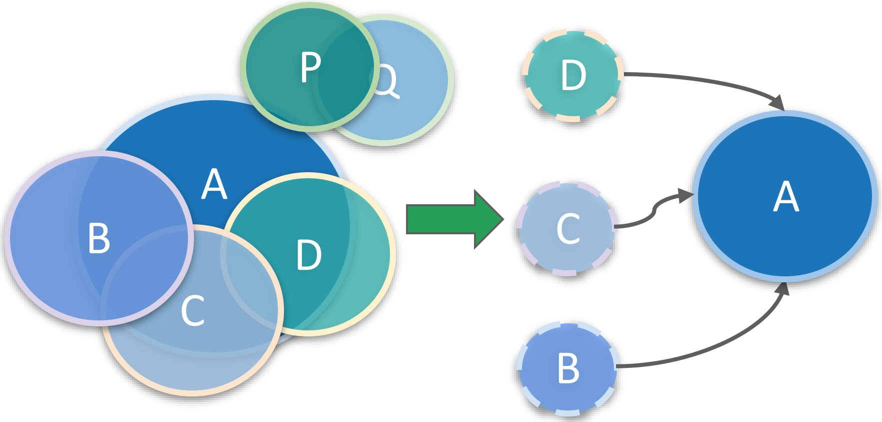
Isolate → Share → Reuse

Consolidation Approaches

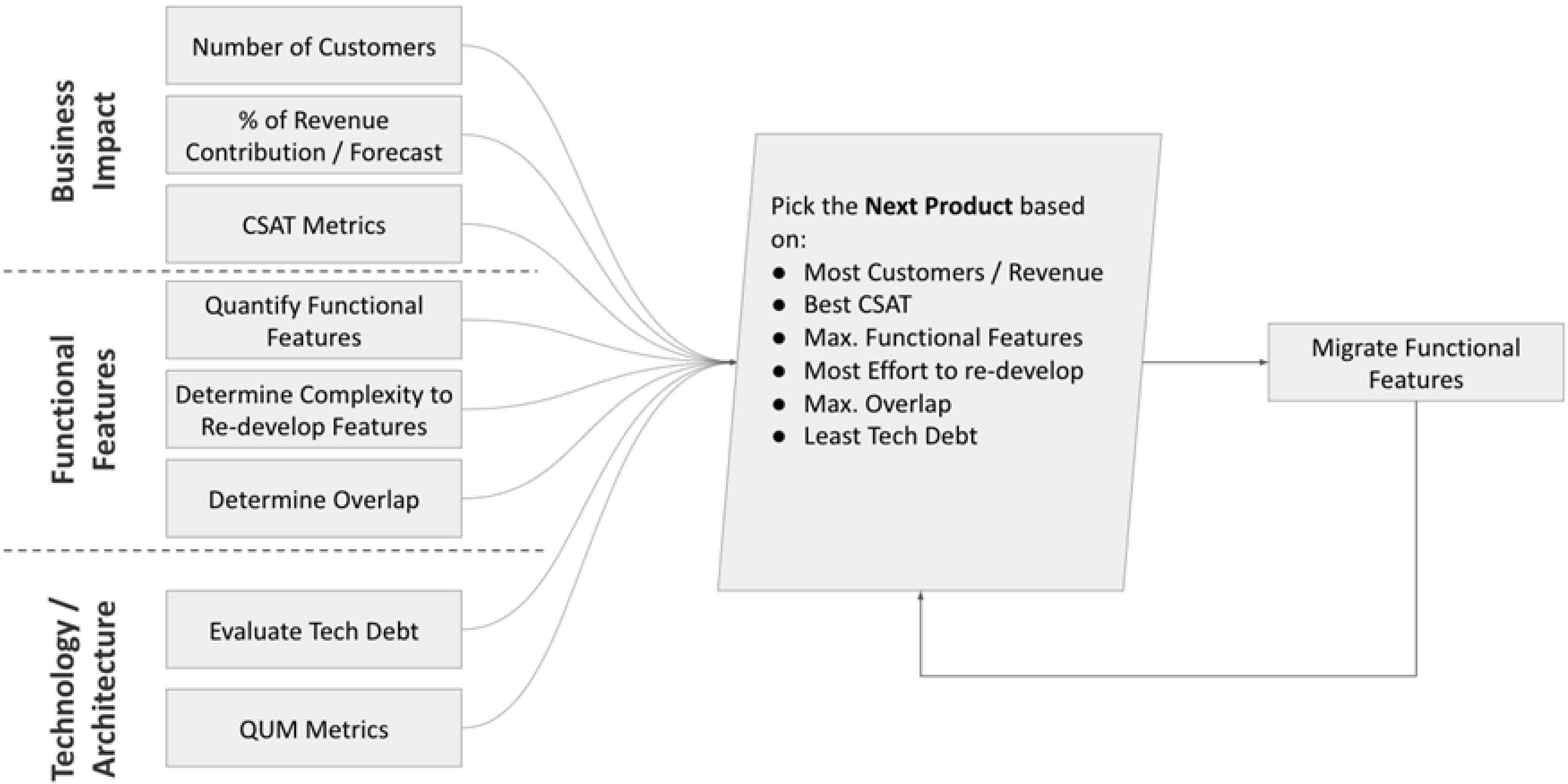
Develop New Unified Product



Nominate & Enhance Existing Product



Iterative Consolidation Process



Business Impact

- Number of Customers
- % of Revenue Contribution / Forecast
- CSAT Metrics

Functional Features

- Quantify Functional Features
- Determine Complexity to Re-develop Features
- Determine Overlap

Technology / Architecture

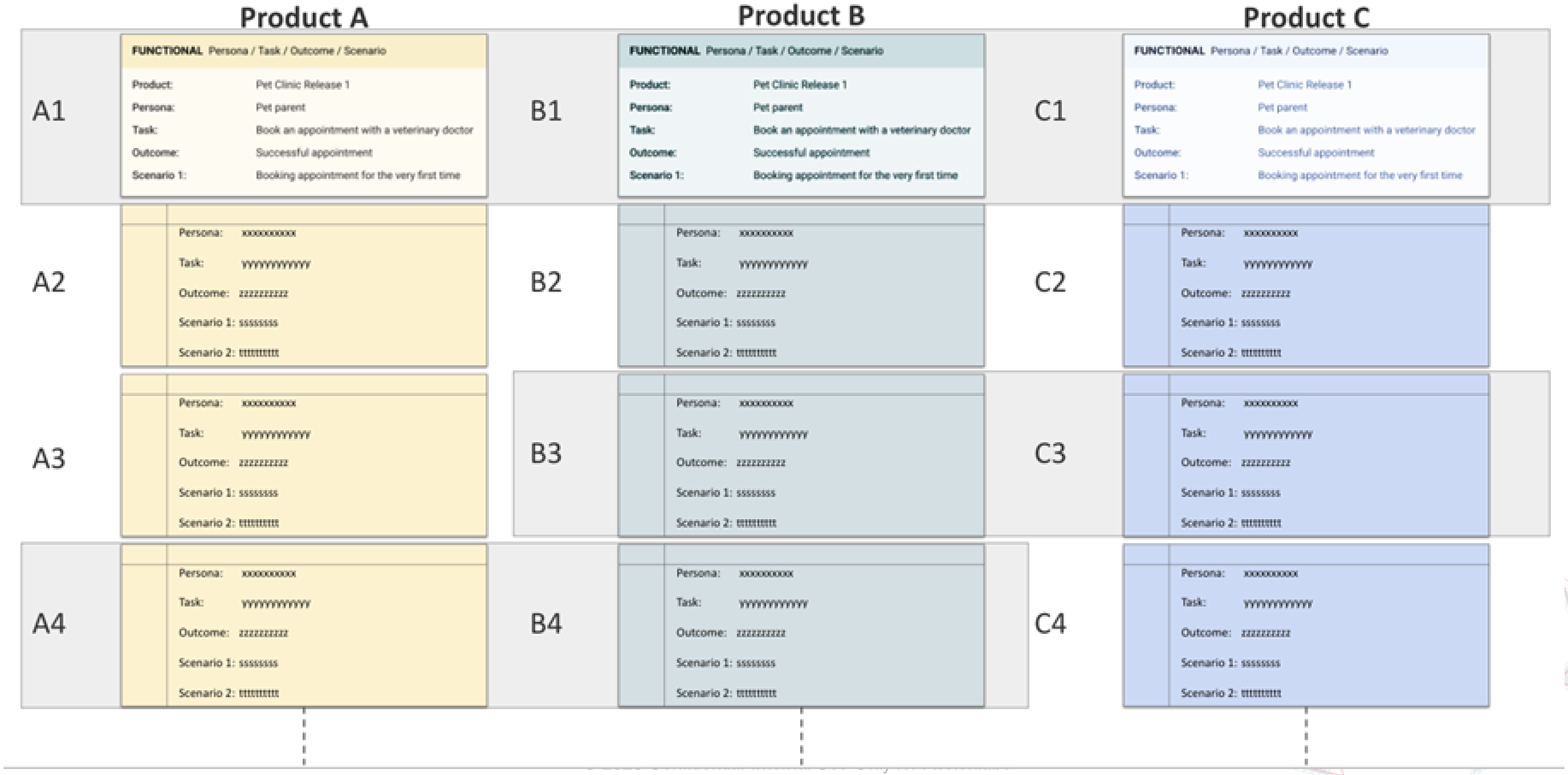
- Evaluate Tech Debt
- QUM Metrics

Pick the **Next Product** based on:

- Most Customers / Revenue
- Best CSAT
- Max. Functional Features
- Most Effort to re-develop
- Max. Overlap
- Least Tech Debt

Migrate Functional Features

QUM Methodology to Measure Functional Overlap



Product A

A1

FUNCTIONAL Persona / Task / Outcome / Scenario

Product: Pet Clinic Release 1
Persona: Pet parent
Task: Book an appointment with a veterinary doctor
Outcome: Successful appointment
Scenario 1: Booking appointment for the very first time

A2

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

A3

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

A4

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

Product B

B1

FUNCTIONAL Persona / Task / Outcome / Scenario

Product: Pet Clinic Release 1
Persona: Pet parent
Task: Book an appointment with a veterinary doctor
Outcome: Successful appointment
Scenario 1: Booking appointment for the very first time

B2

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

B3

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

B4

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

Product C

C1

FUNCTIONAL Persona / Task / Outcome / Scenario

Product: Pet Clinic Release 1
Persona: Pet parent
Task: Book an appointment with a veterinary doctor
Outcome: Successful appointment
Scenario 1: Booking appointment for the very first time

C2

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

C3

Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

C4

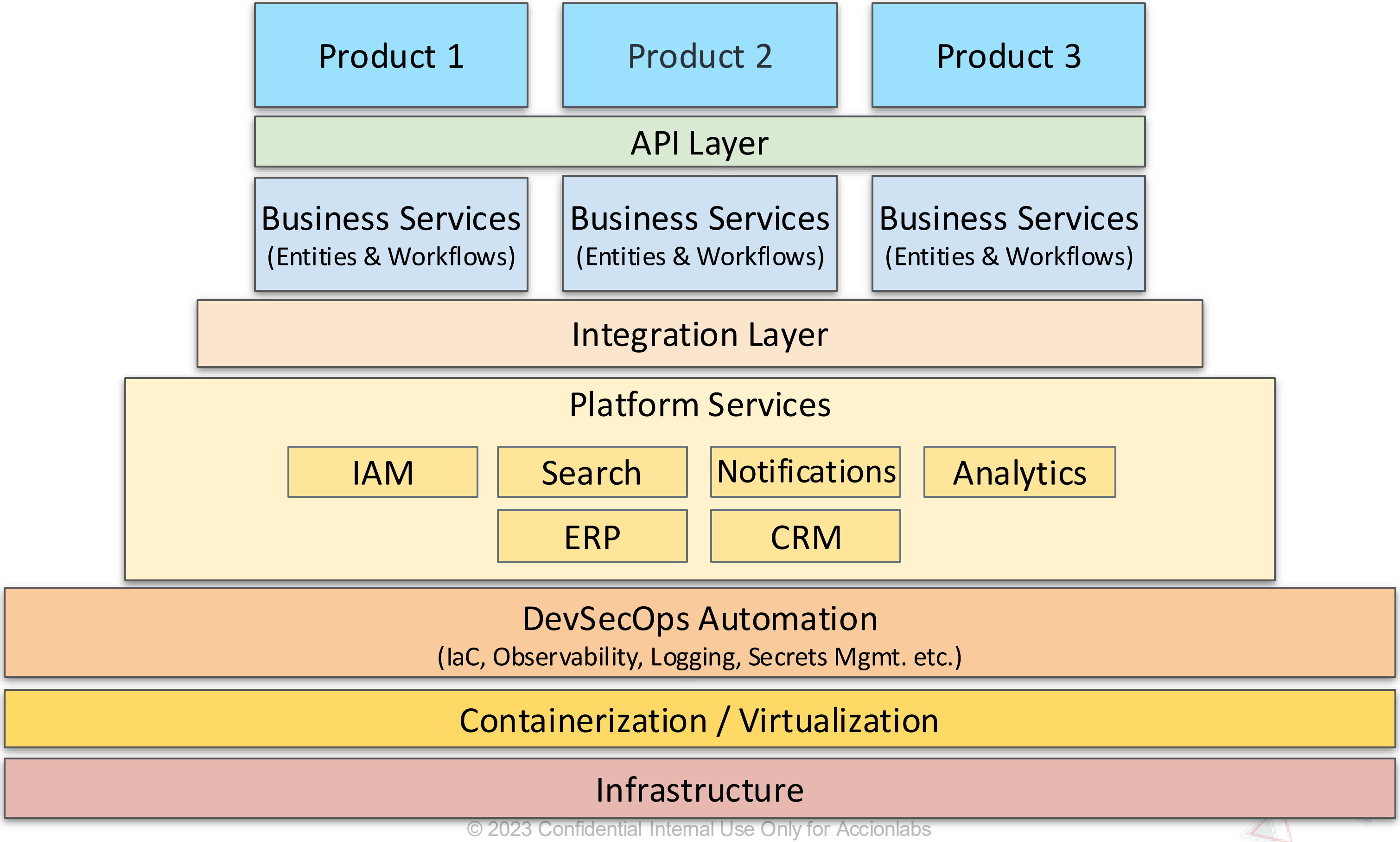
Persona: xxxxxxxxxx
Task: yyyyyyyyyyy
Outcome: zzzzzzzzzz
Scenario 1: ssssssss
Scenario 2: tttttttttt

Options to Bridge Functional Gaps

- Merge code from separate code bases
  - Migration of code from one product to another.
  - Applicable only for same / similar tech stacks
  - This may not be always possible / likely to be the case.
- Create Shared libraries / Reusable components
  - Create shared libraries / reusable components that can be reused
  - Applicable only for same / similar tech stacks
  - Applicable in scenarios where APIs are not available
- Share Backend via API Integrations
  - Applicable for polyglot products when APIs are available.
  - May require additional considerations like SSO, synchronizing master data and other data elements may be necessary.
- Share Backend via Microservices
  - Cull out Microservice from Product A to Product B
  - Applicable for polyglot products when the coupling between the implementation of features is minimal or none.
- Share Backend + Frontend - Microservices + Micro Frontends
  - Cull out the Front end in the form of micro front end and Microservice from Product A to Product B
- Write Code from Scratch
  - Least ideal approach, but consider in worst case scenario.

Architectural Overlap - Isolate, Share & Reuse

Accionlabs



Product 1

Product 2

Product 3

API Layer

Business Services
(Entities & Workflows)

Business Services
(Entities & Workflows)

Business Services
(Entities & Workflows)

Integration Layer

Platform Services

IAM

Search

Notifications

Analytics

ERP

CRM

DevSecOps Automation
(IaC, Observability, Logging, Secrets Mgmt. etc.)

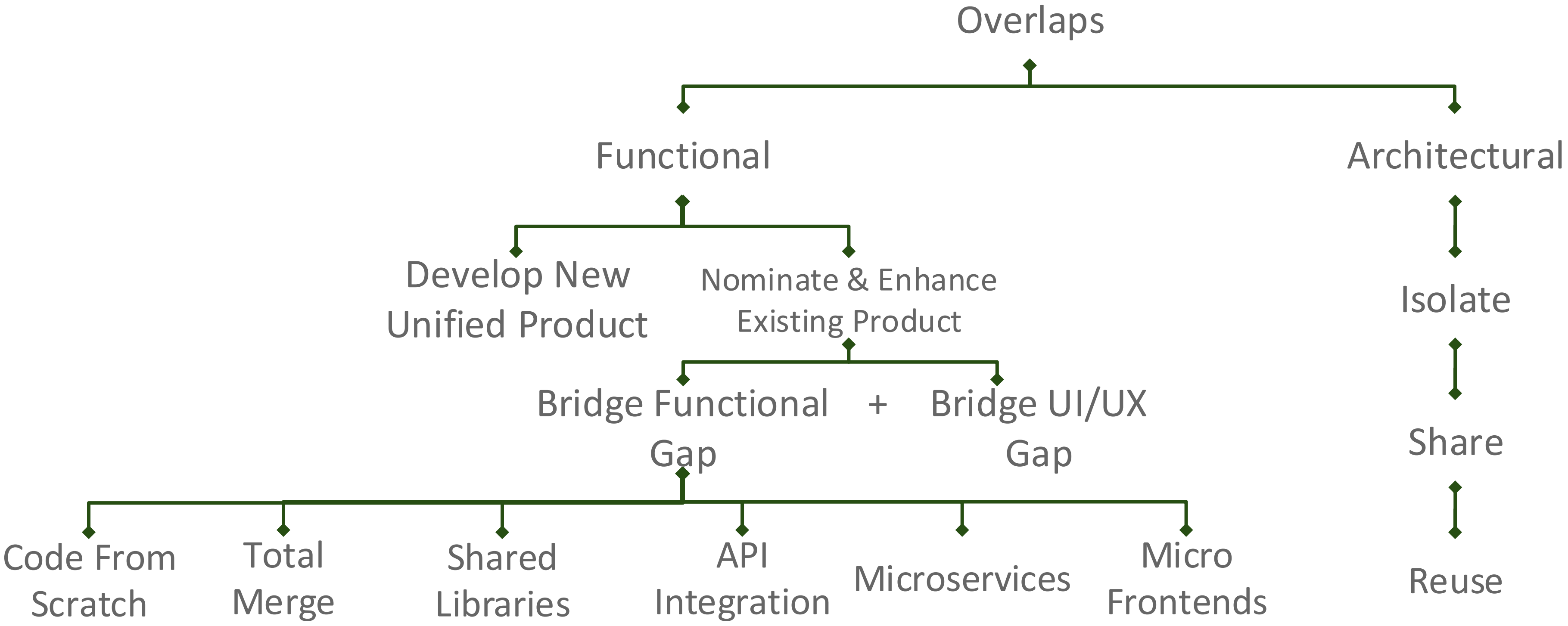
Containerization / Virtualization

Infrastructure

Key Considerations in Rationalization

Category	Area
User Experience	Adaptability
	Feature Parity
	Internationalization / Localization
Business Process	Workflows and Business Rules Consolidation / Simplification
	Automation
Technology	Current Tech Debt
	Data Quality / MDM
	Data Migration
People	Customer Migration
	Team Skills
Economics	Capital Investments
	Operational Costs
	Time to Market, ROI

Summary



Overlaps

Functional

Architectural

Develop New Unified Product

Nominate & Enhance Existing Product

Bridge Functional Gap + Bridge UI/UX Gap

Code From Scratch

Total Merge

Shared Libraries

API Integration

Microservices

Micro Frontends

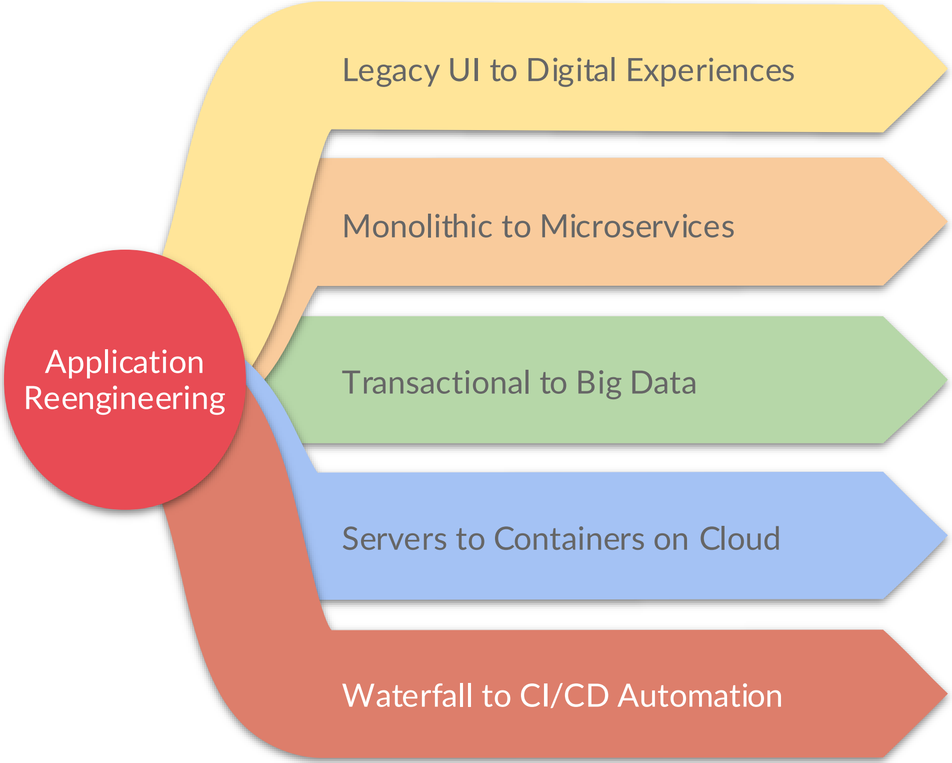
Isolate

Share

Reuse

Re-Engineering Methodology

Reengineering / Modernization Work Streams

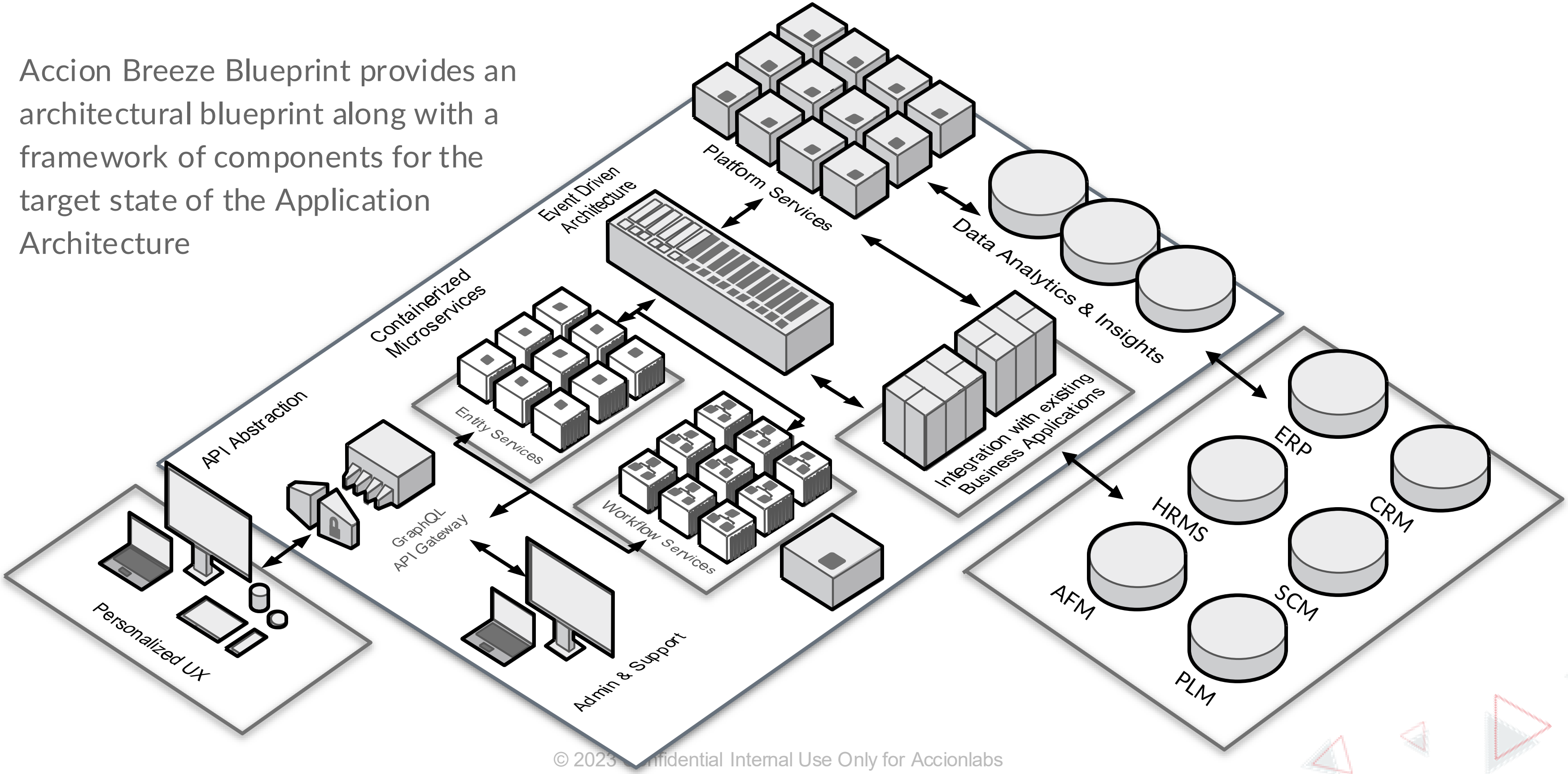

Application Reengineering
Legacy UI to Digital Experiences
Monolithic to Microservices
Transactional to Big Data
Servers to Containers on Cloud
Waterfall to CI/CD Automation

- Accion helps clients to re-engineer their legacy software applications into scalable, high performing Digital Products
- Multiple re-engineering work streams are prioritized and executed in parallel
- Applications are re-engineered in iterative releases, so that customers can adopt new products in stages
- Accion helps to develop, maintain and implement the re-engineered digital products
- Accion Breeze provides a comprehensive foundation for building digital platforms

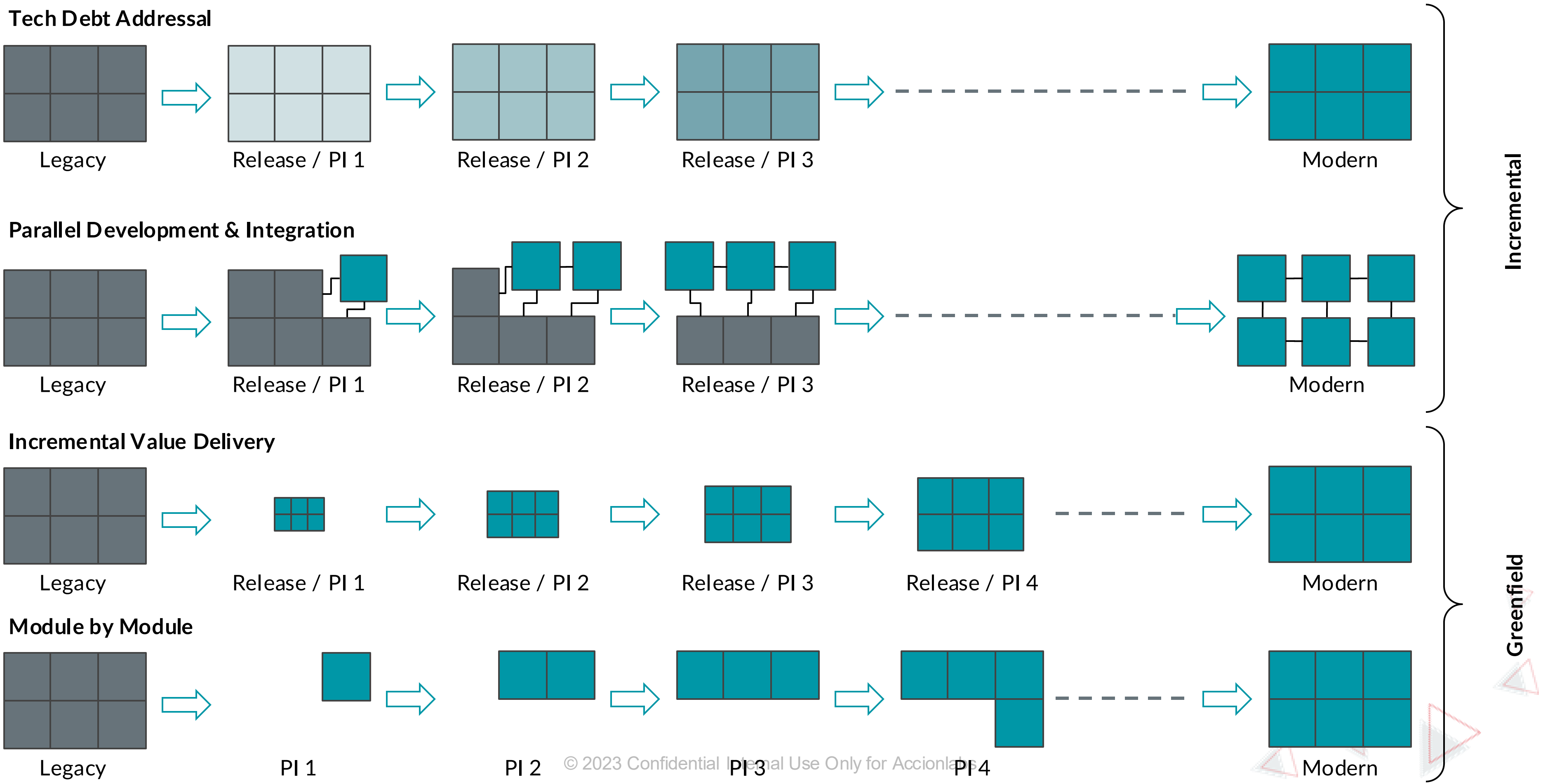
Breeze - Digital Architecture Blueprint

Accionlabs

Accion Breeze Blueprint provides an architectural blueprint along with a framework of components for the target state of the Application Architecture

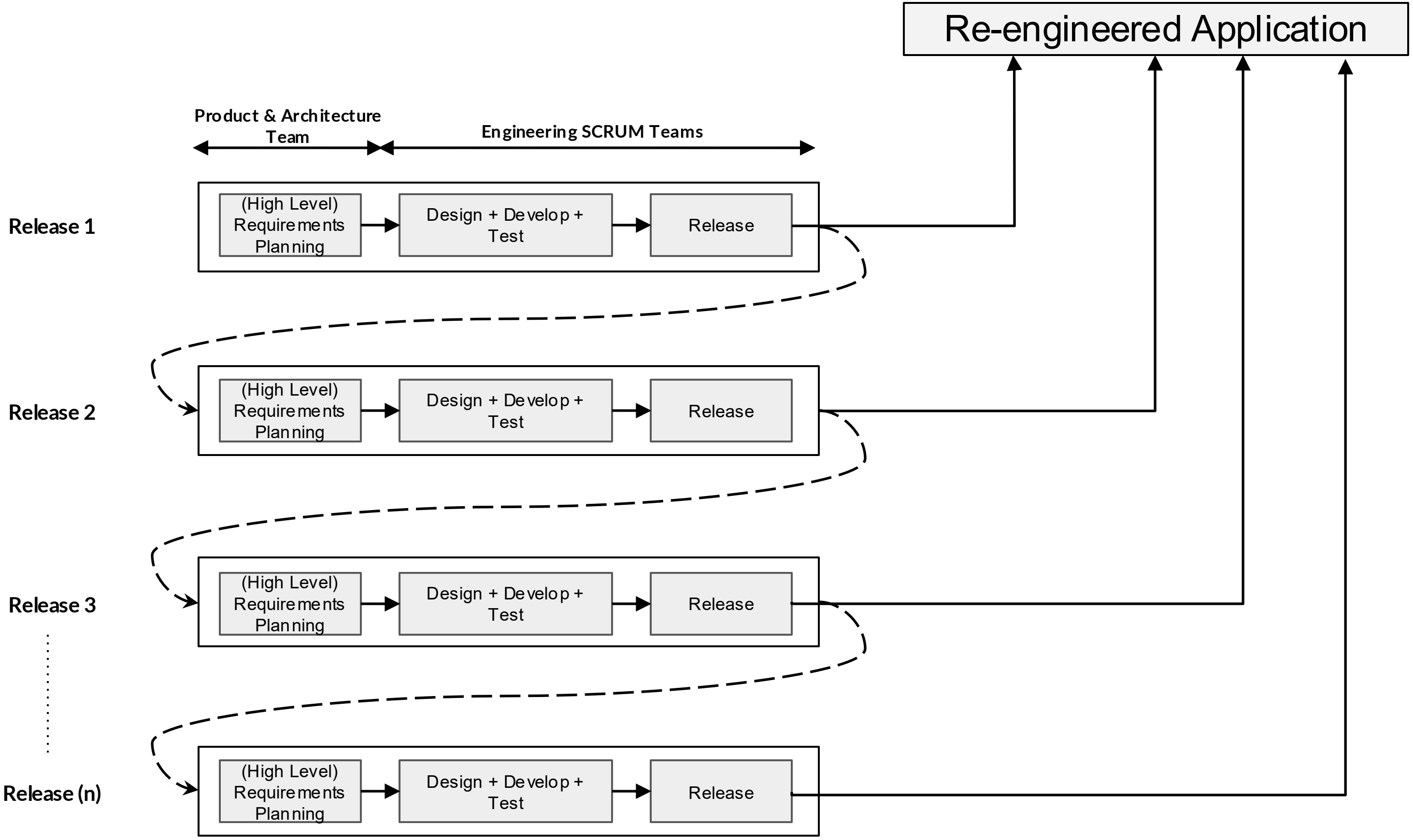


Re-Engineering Approaches



Time Boxed Release Methodology

Accionlabs


Re-engineered Application
Product & Architecture Team
Engineering SCRUM Teams
Release 1
(High Level) Requirements Planning
Design + Develop + Test
Release
Release 2
(High Level) Requirements Planning
Design + Develop + Test
Release
Release 3
(High Level) Requirements Planning
Design + Develop + Test
Release
Release (n)
(High Level) Requirements Planning
Design + Develop + Test
Release

Case Studies

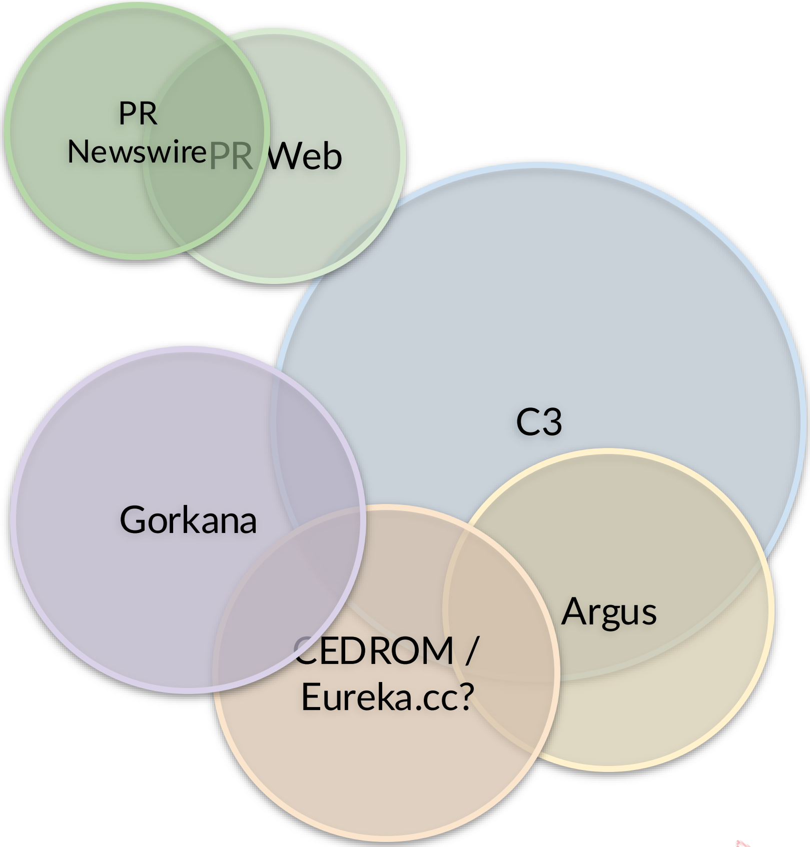
MarTech Product Portfolio Analysis

Metrics	C3	Argus	CEDROM	Gorkana	PRWeb	PRNewswire
Category	Media Monitoring, Insights and Analysis, Targeting				Content Creation, Scheduling, Distribution, Tracking and Analysis	
Markets	USA / Worldwide	France	Canada & France	UK	USA / Worldwide	USA / Worldwide
Media Type	Digital Media	Digital Media / Traditional Media?	Digital and Traditional Media	Digital Media	Digital Media, Press Releases, Digital Assets	Digital Media, Press Releases, Digital Assets
Competitors	Meltwater, Mention, Critical Mention, Hootsuite, Sprout Social, Muck Rack				Businesswire, Newswire, Globe Newswire	
History / Origin	Homegrown, Flagship Product	Acquired from Argus de la presse	Acquired from CEDROM CNi	Acquired from Gorkana Group	Homegrown	Homegrown

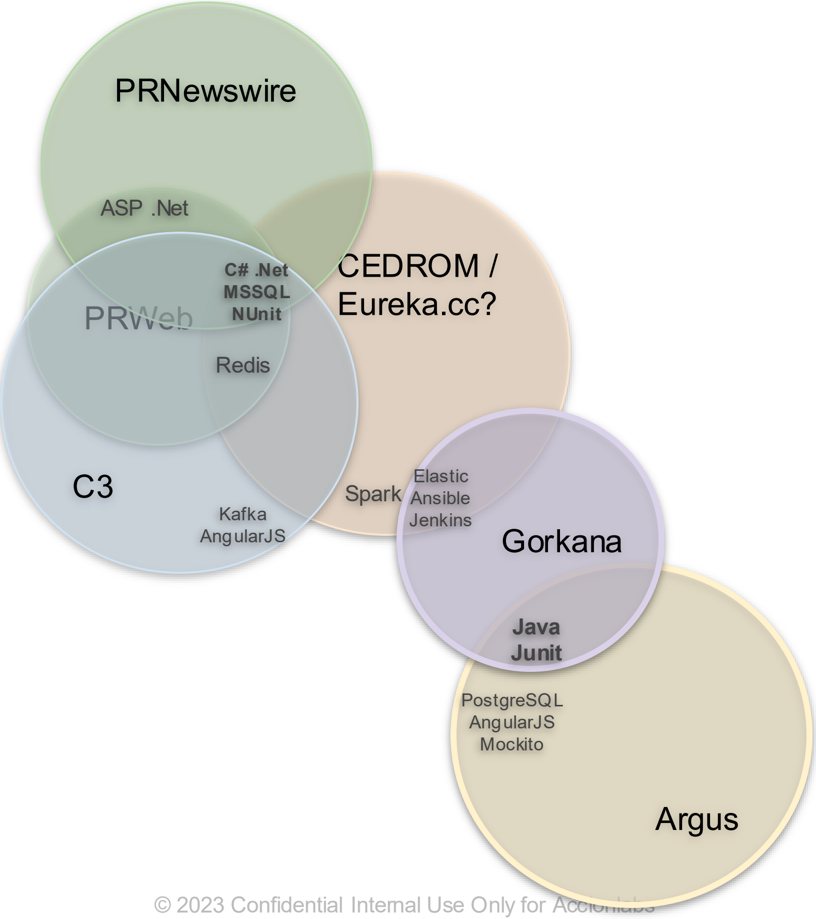
Monitoring - Product Features Overlap

Distinguishing Factors:

- Primary features of the product
- Distinct Media Types for Analysis
  - TV media
  - Radio
  - Podcasts
  - Newspapers
  - Magazines
  - Social Media etc.
- Geo specific data sources
- Locale coverage of contacts / influencers
- Dependencies on other applications:
  - LuQi
  - Visible
  - IRIS
  - WISE etc.

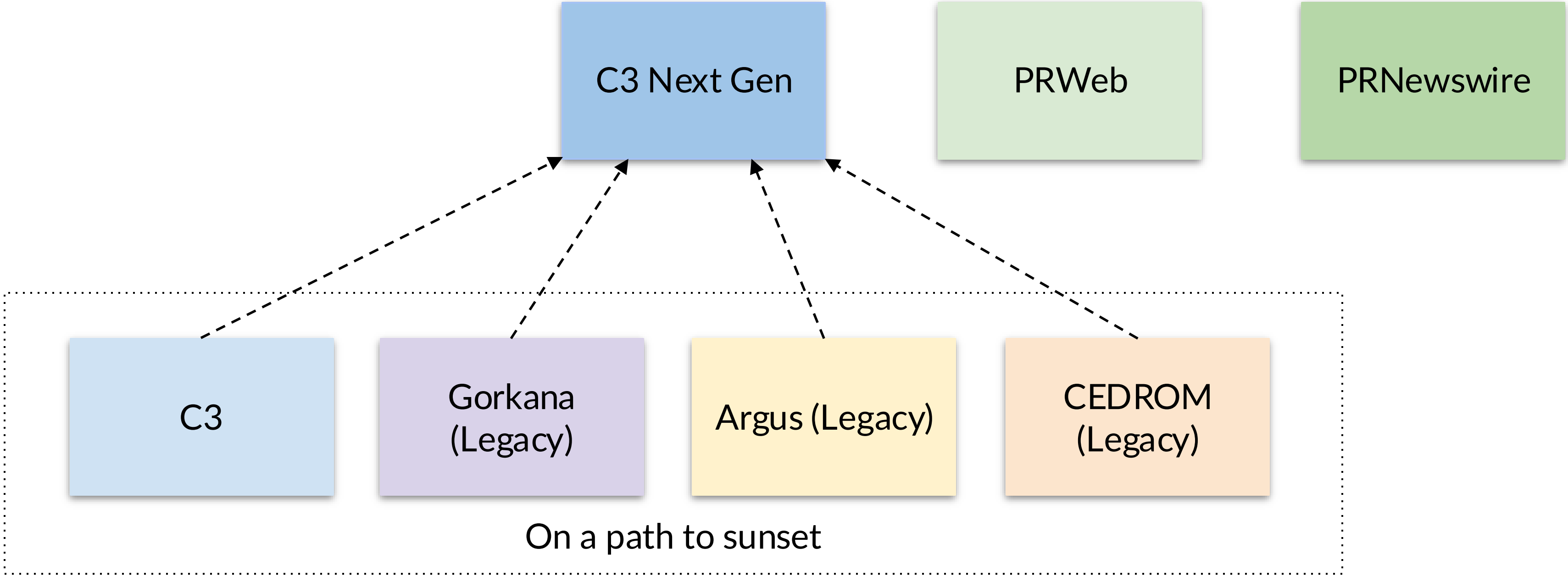


Monitoring - Product Technologies Overlap



Product Portfolio Consolidation + Reengineering

Accionlabs



PR Distribution Platform Reengineering

Legacy Architecture

- Custom Built Monolithic Legacy Systems
  - Multiple Technology Stacks
  - Duplicate Features in multiple applications
  - Difficult to add new features
  - Third party components with no support
  - Scalability and performance issues
  - Silo databases with complex synchronization
- Legacy UI / UX
  - Legacy UI architecture (ASP.net)
  - Multiple front end portals with duplicate features
  - No mobile support
  - Custom front end for large customers
- Legacy Infrastructure
  - On Premise Data Center
  - Legacy Infrastructure Investments
  - Internal IT Support
  - Performance and Scalability Issues

Problem Areas

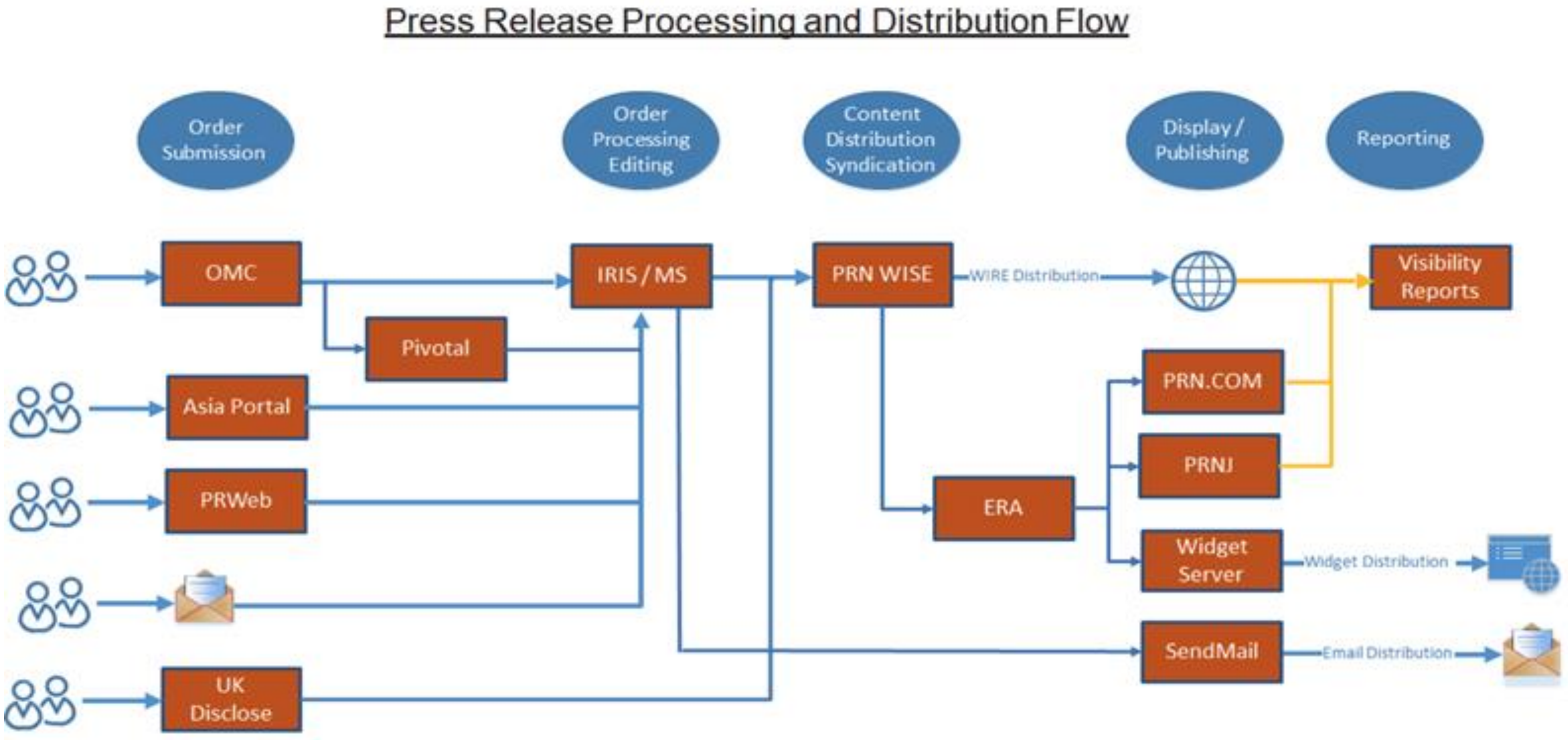
- High cost of maintenance for legacy systems
  - Each backend system has dedicated teams
  - Multiple tech skills and architecture
  - Undocumented code and APIs
  - Frequent bug fixes
- Ineffective Customer Experience
  - Different front end for different geographies
  - No path for cross sales or feature parity
  - Competition from new players with modern apps
  - Frequent customer complaints on UX
- Brittle Infrastructure
  - Duplicate data in multiple systems leading to high data management costs
  - Legacy investments in IT infrastructure
  - High internal costs
  - No easy path for cloud migration

Reengineering Objectives

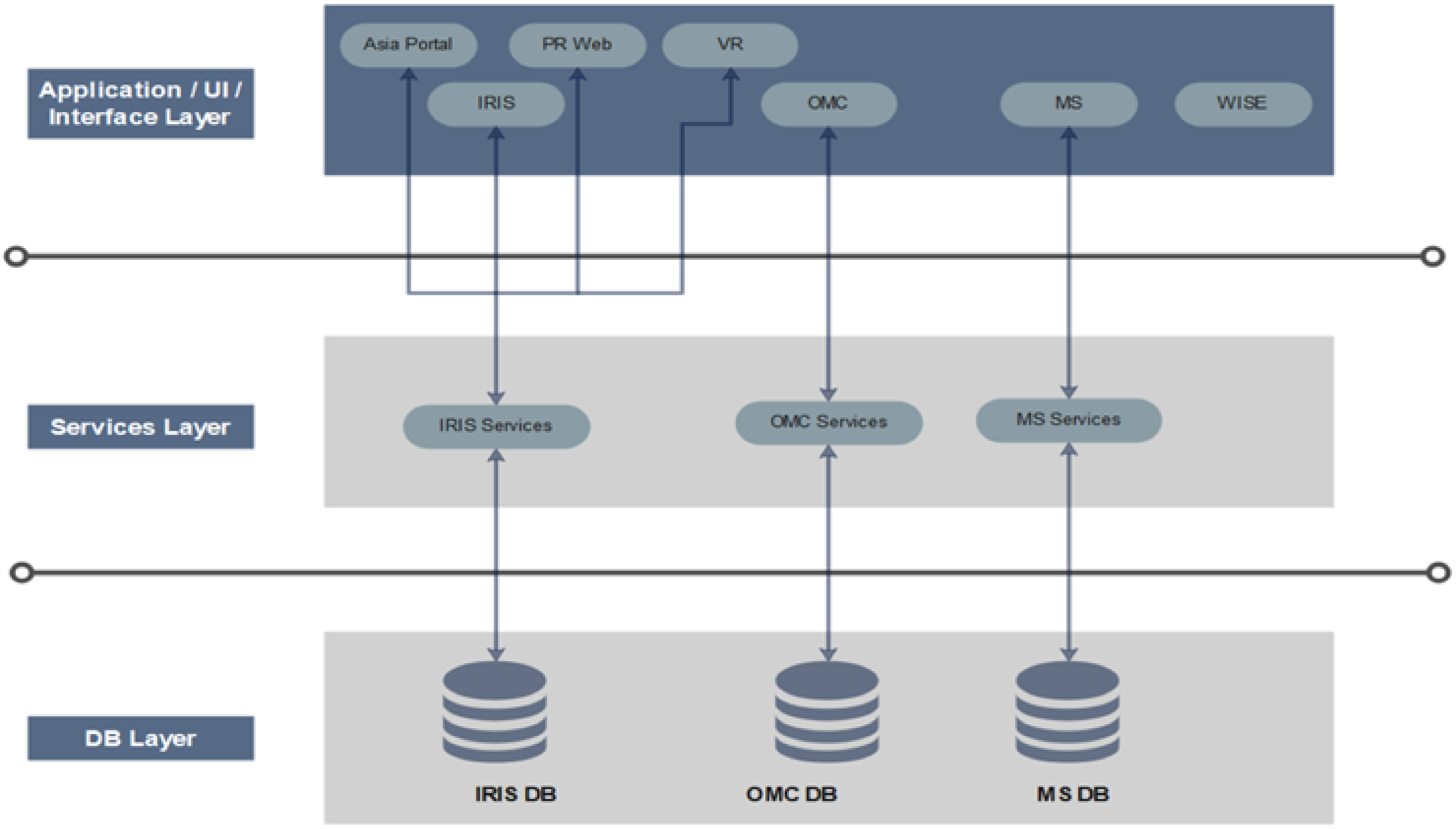
- Single Technology Stack
  - Multiple products through acquisitions and internal growth, with different technology stack. Moving to a single technology stack will increase reusability and reduce maintenance overhead
- High Availability
  - All systems should be available 24/7/365 except planned down times. System monitoring needs to expose the health of all components.
- Easy and Flexible Deployment
  - New features should be easy to deploy without compromising dependencies between components
- Modernize User Experience
  - Personalized user interfaces that recognize and customize the views for each individual user, and allow innovations to surface easily
- Support Future Business Requirements
  - The reengineered architecture should support making significant changes based on new requirements from customers, markets or competitors

- Ease of Integration
  - It should be easy to integrate other external or internal applications with adequate measures of security and access control implemented
- Security Compliance
  - The application architecture needs to be compliant with all security and privacy requirements as mandated by customers or internal guidelines
- Seamless Patching or Rebuilding
  - It should be possible to seamlessly patch in new versions, bug fixes or data issues with no disruption to the smooth functioning of the application

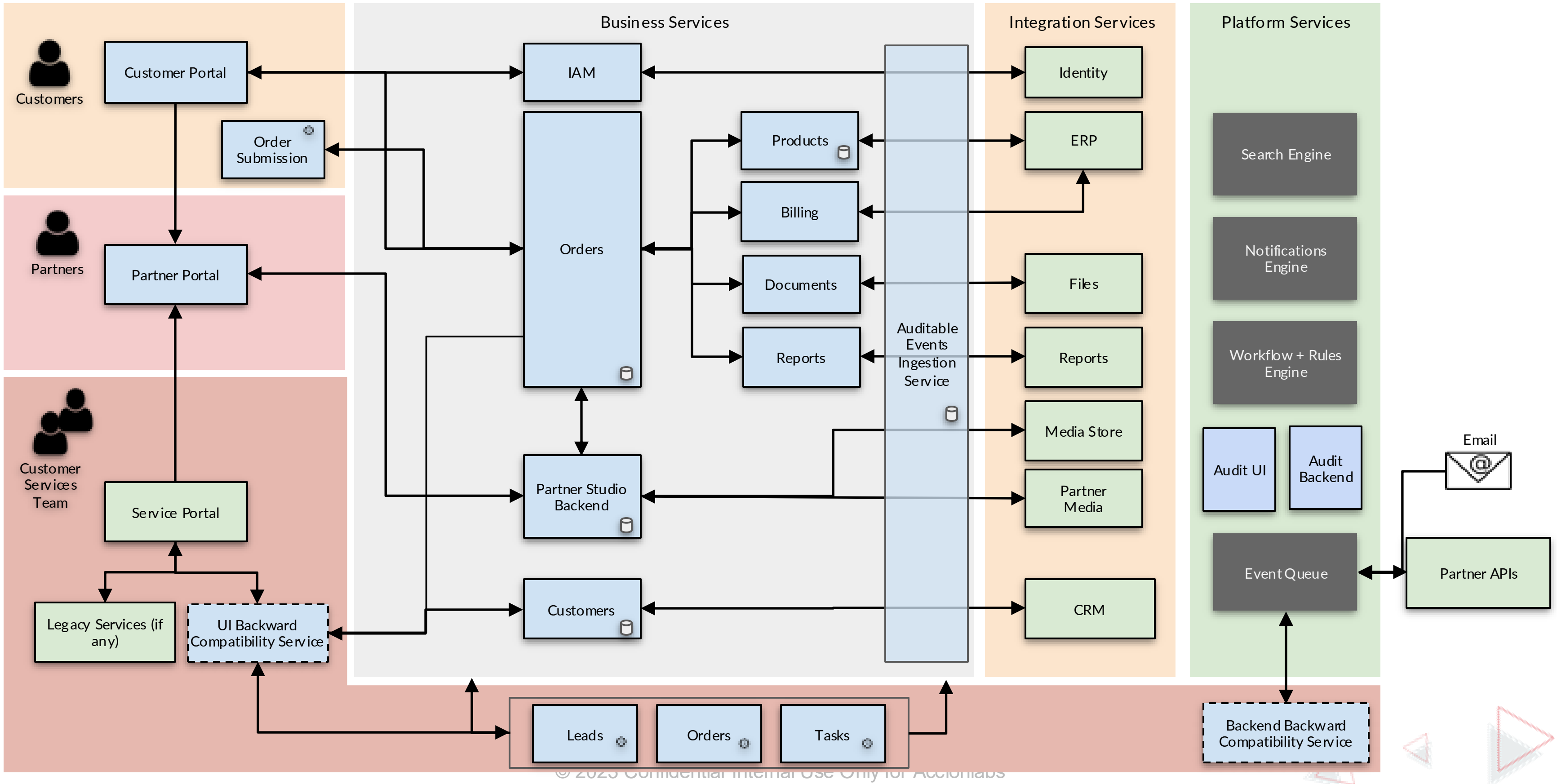
Distribution - Functional Analysis



Distribution - Technology / Architecture Analysis

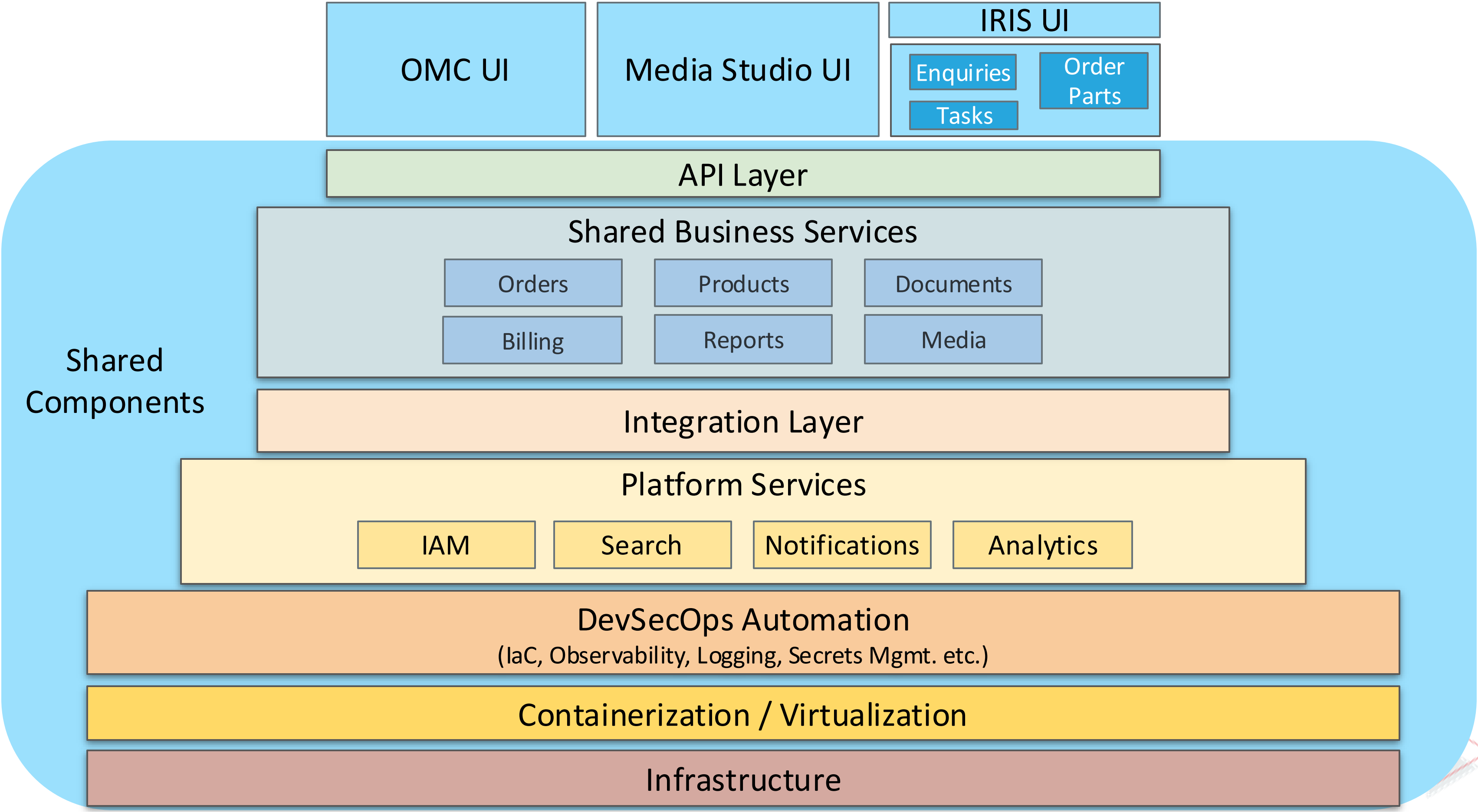


Distribution - Solution Conceptual Architecture



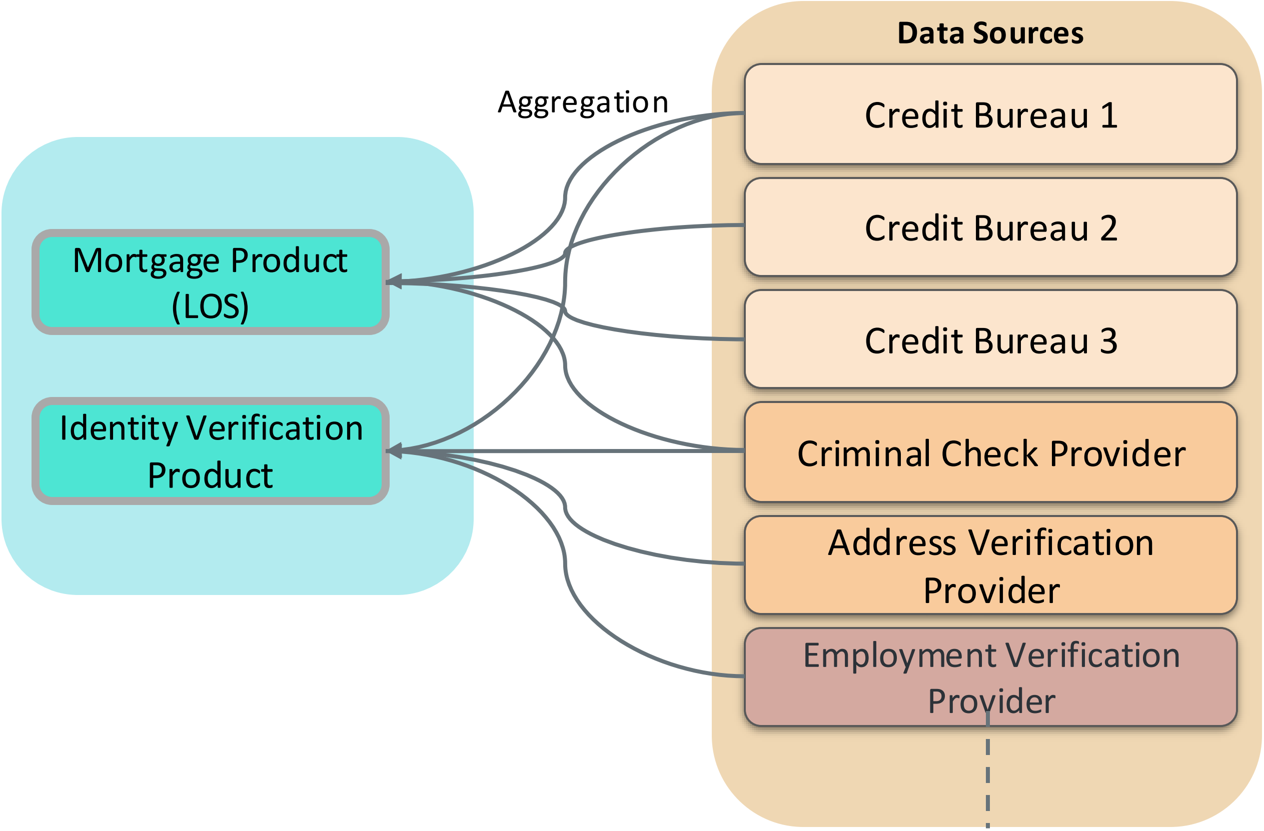
Distribution - Architecture Overlap

Accionlabs



Credit Aggregator - Product Portfolio

Accionlabs

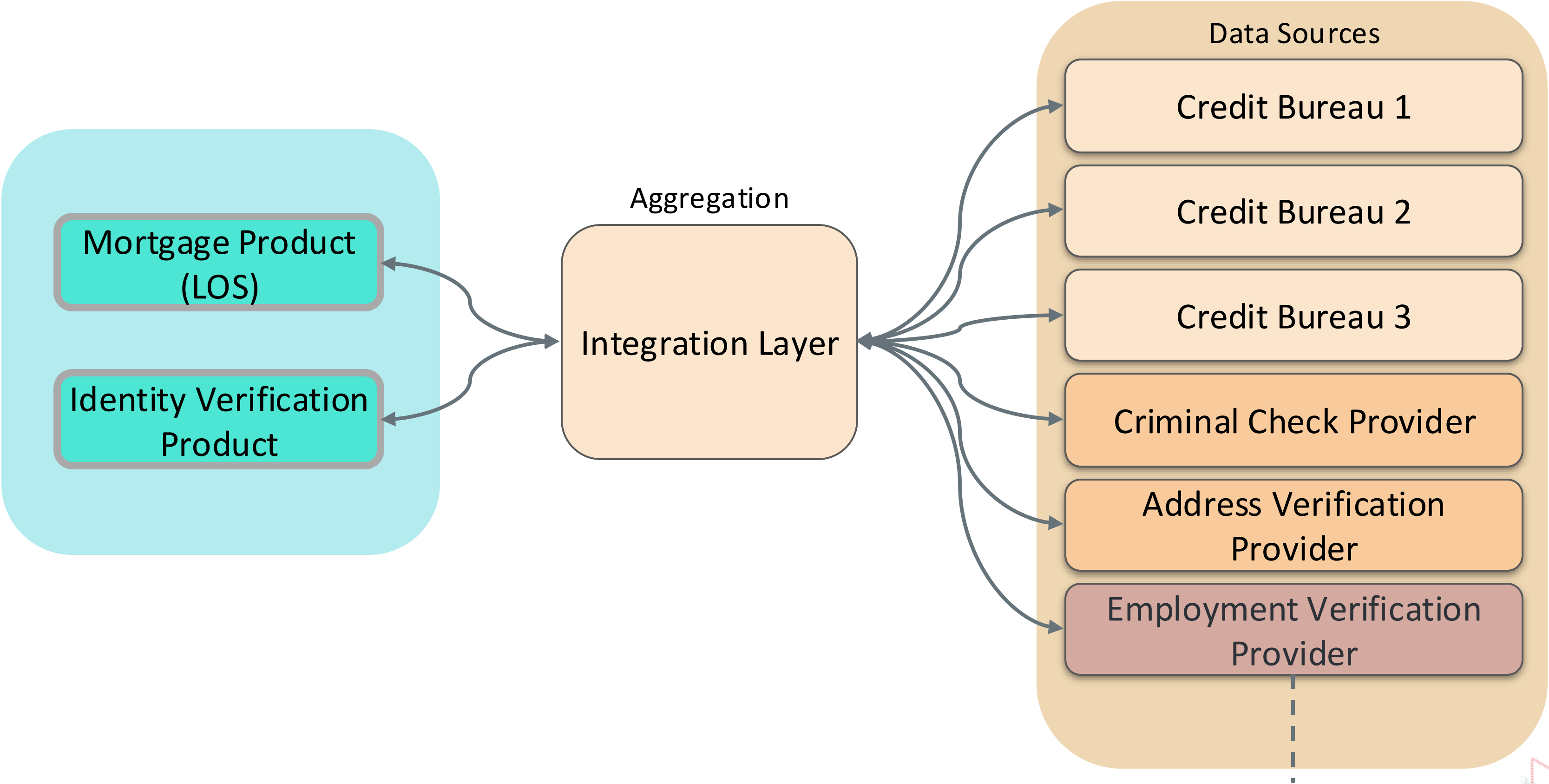


Credit Aggregator

Motivation: Additional Revenue Streams & Adjunct Innovation
Methodology: Self Developed

Credit Aggregator - Solution Architecture

Accionlabs



Thank You!