AGILE DELIVERY FOR SCALABLE SUPPLY CHAIN PLATFORM

Case Study By:
Sumeet Boob

AGENDA

- Business Context & Objectives
- Solution Scope & NFRs
- Target Architecture
- Agile Delivery Strategy
- Delivery Organization & Resource Model
- Estimation & Capacity Planning
- Engineering & Delivery Framework
- Governance, Risks & Assumptions
- Engagement Model
- Key Takeaways

CONTEXT & OBJECTIVES

Objective:
Design a scalable digital supply chain platform enabling suppliers, operators, and logistics partners to manage inventory, orders, and deliveries with real-time visibility.

In Scope

- Supply chain platform (web & mobile)
- Inventory & order management
- Supplier integration
- Analytics dashboards

Out of Scope

- Supplier ERP changes
- Logistics internal systems
- Hardware/infrastructure changes

SCOPE & NON-FUNCTIONAL REQUIREMENTS

Functional Scope

The platform will deliver **an end-to-end digital solution for managing supplier onboarding, inventory visibility, order processing, and delivery tracking across web and mobile channels**, allowing businesses to oversee inventory, orders, and deliveries via both web and mobile interfaces.

Authorized business users will have the capability to **initiate and oversee orders with multiple third-party suppliers**, covering the entire order lifecycle within the platform.

The system will **streamline inventory control, order processing, and product tracking**, offering real-time insights into product availability and delivery progress.

A data platform will facilitate the **gathering, analysis, and reporting of operational data** to enhance demand and supply planning, support decision-making, and improve operational efficiency.

Additionally, the solution will provide **near real-time monitoring and operational analytics**, empowering production teams to effectively oversee platform health and perf.

Non-Functional Requirements

Performance – API response time < **2 seconds** under normal load; validated through performance testing.

Usability – **Responsive web and mobile interfaces** supporting consistent experience across devices.

Security – **RBAC-based access control** with complete **audit trail** for all data operations.

Availability – Platform designed for **99.99% uptime SLA** with continuous access during business operations.

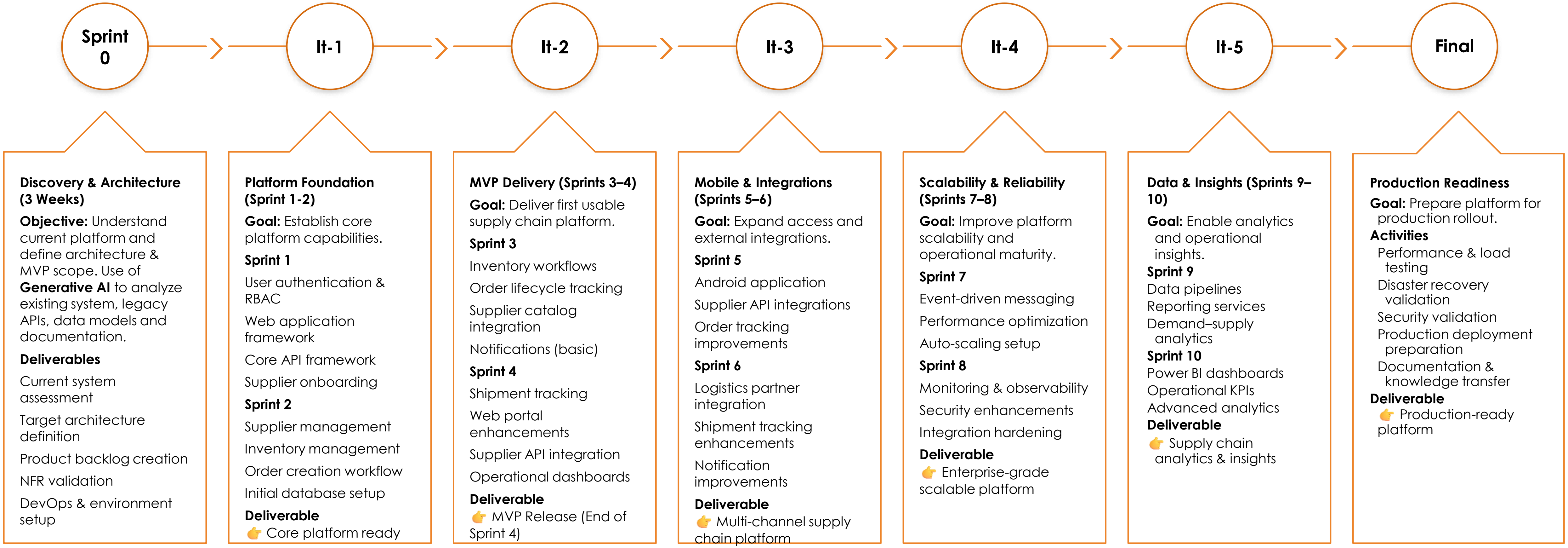
Scalability – **Auto-scaling services** to handle varying workloads without manual intervention.

Maintainability – **Centralized logging and diagnostics** to enable faster issue identification and resolution.

Supportability – **Real-time monitoring and alerts** for application and infrastructure health.

Extensibility – **Modular architecture** allowing new services or integrations without impacting existing functionality.

AGILE DELIVERY PLAN



Discovery & Architecture (3 Weeks)

Objective: Understand current platform and define architecture & MVP scope. Use of **Generative AI** to analyze existing system, legacy APIs, data models and documentation.

Deliverables

Current system assessment

Target architecture definition

Product backlog creation

NFR validation

DevOps & environment setup

Platform Foundation (Sprint 1-2)

Goal: Establish core platform capabilities.

Sprint 1

User authentication & RBAC

Web application framework

Core API framework

Supplier onboarding

Sprint 2

Supplier management

Inventory management

Order creation workflow

Initial database setup

Deliverable

👉 Core platform ready

MVP Delivery (Sprints 3–4)

Goal: Deliver first usable supply chain platform.

Sprint 3

Inventory workflows

Order lifecycle tracking

Supplier catalog integration

Notifications (basic)

Sprint 4

Shipment tracking

Web portal enhancements

Supplier API integration

Operational dashboards

Deliverable

👉 MVP Release (End of Sprint 4)

Mobile & Integrations (Sprints 5–6)

Goal: Expand access and external integrations.

Sprint 5

Android application

Supplier API integrations

Order tracking improvements

Sprint 6

Logistics partner integration

Shipment tracking enhancements

Notification improvements

Deliverable

👉 Multi-channel supply chain platform

Scalability & Reliability (Sprints 7–8)

Goal: Improve platform scalability and operational maturity.

Sprint 7

Event-driven messaging

Performance optimization

Auto-scaling setup

Sprint 8

Monitoring & observability

Security enhancements

Integration hardening

Deliverable

👉 Enterprise-grade scalable platform

Data & Insights (Sprints 9–10)

Goal: Enable analytics and operational insights.

Sprint 9

Data pipelines

Reporting services

Demand–supply analytics

Sprint 10

Power BI dashboards

Operational KPIs

Advanced analytics

Deliverable

👉 Supply chain analytics & insights

Production Readiness

Goal: Prepare platform for production rollout.

Activities

Performance & load testing

Disaster recovery validation

Security validation

Production deployment preparation

Documentation & knowledge transfer

Deliverable

👉 Production-ready platform

SOLUTION ARCHITECTURE OVERVIEW

1️⃣ Experience Layer – Multi-Channel Access

Platform supports:
- **iOS (Swift)**
- **Android (Kotlin)**
- **Web Portal (ReactJS)**

Clients interact through REST APIs ensuring consistent logic and secure access across devices.

👉 **UI remains thin — business logic resides in APIs.**

2️⃣ Secure API Layer (.NET Core)

Central gateway for all client interactions.
- SSO authentication through Azure AD
- Role-based access control
- Request validation and API governance
- Centralized audit logging

Routes validated requests to backend services.

👉 **Controls access, security, and API governance.**

3️⃣ Business Services Layer

Modular domain services handling core supply-chain processes.
- **Inventory Service**
- **Order Management**
- **Supplier Integration**
- **Tracking Service**

Services communicate via standard APIs.

👉 **Business logic centralized and independently scalable.**

4️⃣ Integration Layer

Handles communication with external platforms.
- **Supplier Systems**
- **SAP Cloud**
- **Logistics Platforms**

Connectors isolate external dependencies from internal services.

👉 **External failures do not impact core services.**

5️⃣ Data Platform

Supports operational data and analytics.
- **SQL Server / PostgreSQL – transactional data**
- **Python pipelines – data processing**
- **Power BI dashboards – reporting**

👉 **Operational and analytical workloads separated.**

6️⃣ Cloud & DevOps Foundation (Azure)

Provides the underlying platform capabilities.
- **Azure App Services / Containers**
- **CI/CD pipelines (Azure DevOps)**
- **Monitoring & Observability**
- **Auto-scaling infrastructure**

👉 **Supports SLA, performance, and operational resilience.**

ARCHITECTURE BENEFITS & WORKFLOW

Scalable Microservices
Independent services (Inventory, Orders, Supplier, Tracking) scale without impacting the full platform.

Secure API Access
API Gateway + SSO enable authentication, rate limiting, and controlled access.

Real-Time Visibility
Event-driven messaging supports near real-time inventory and order lifecycle updates.

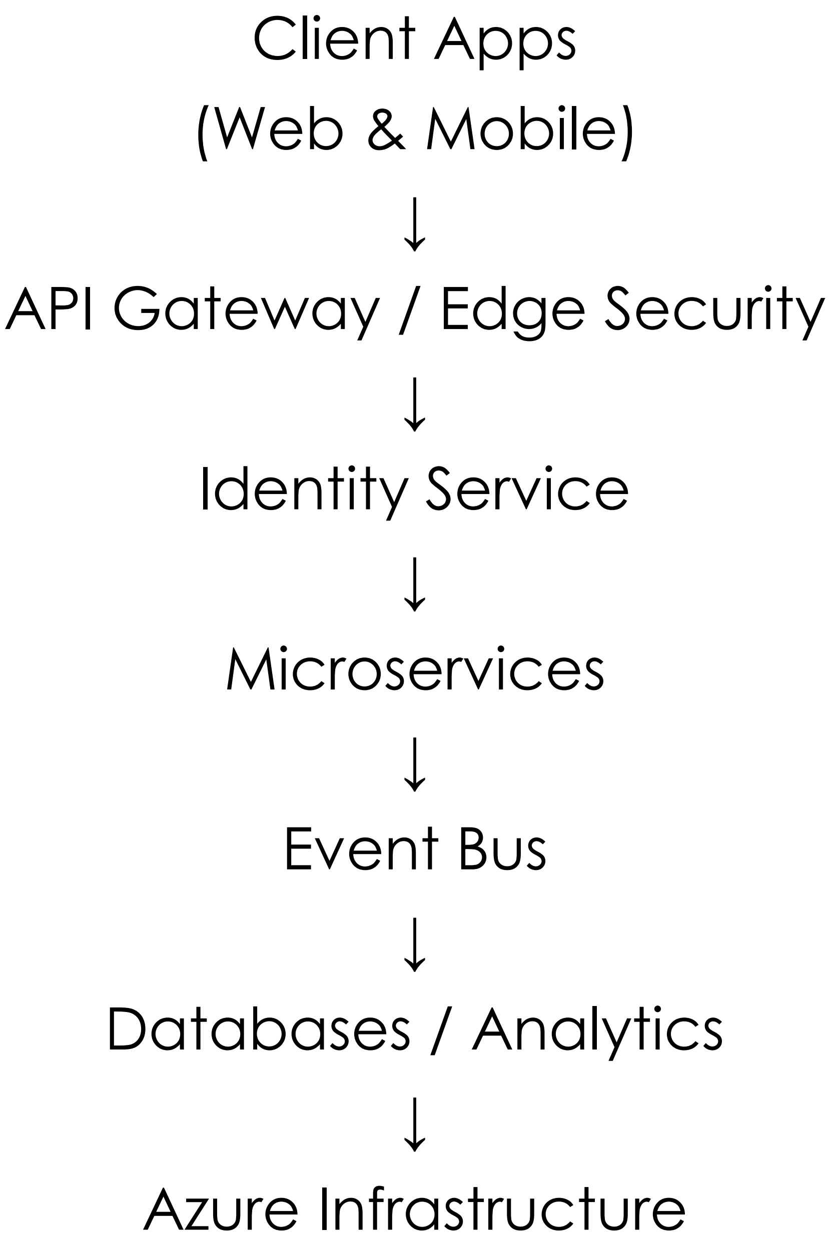
High Availability Architecture
Azure deployment with containerization enables auto-scaling and 99.99% SLA resilience.

Loose External Integrations
Supplier, SAP, and logistics systems isolated through integration services.

Data & Analytics Optimization
Transactional systems separated from analytics pipelines (PowerBI insights).

Operational Observability
Centralized monitoring, logging, and alerting for faster issue resolution.

Extensible Platform Design
API-first modular design enabling easy addition of suppliers and capabilities.



Client Apps
(Web & Mobile)

↓

API Gateway / Edge Security

↓

Identity Service

↓

Microservices

↓

Event Bus

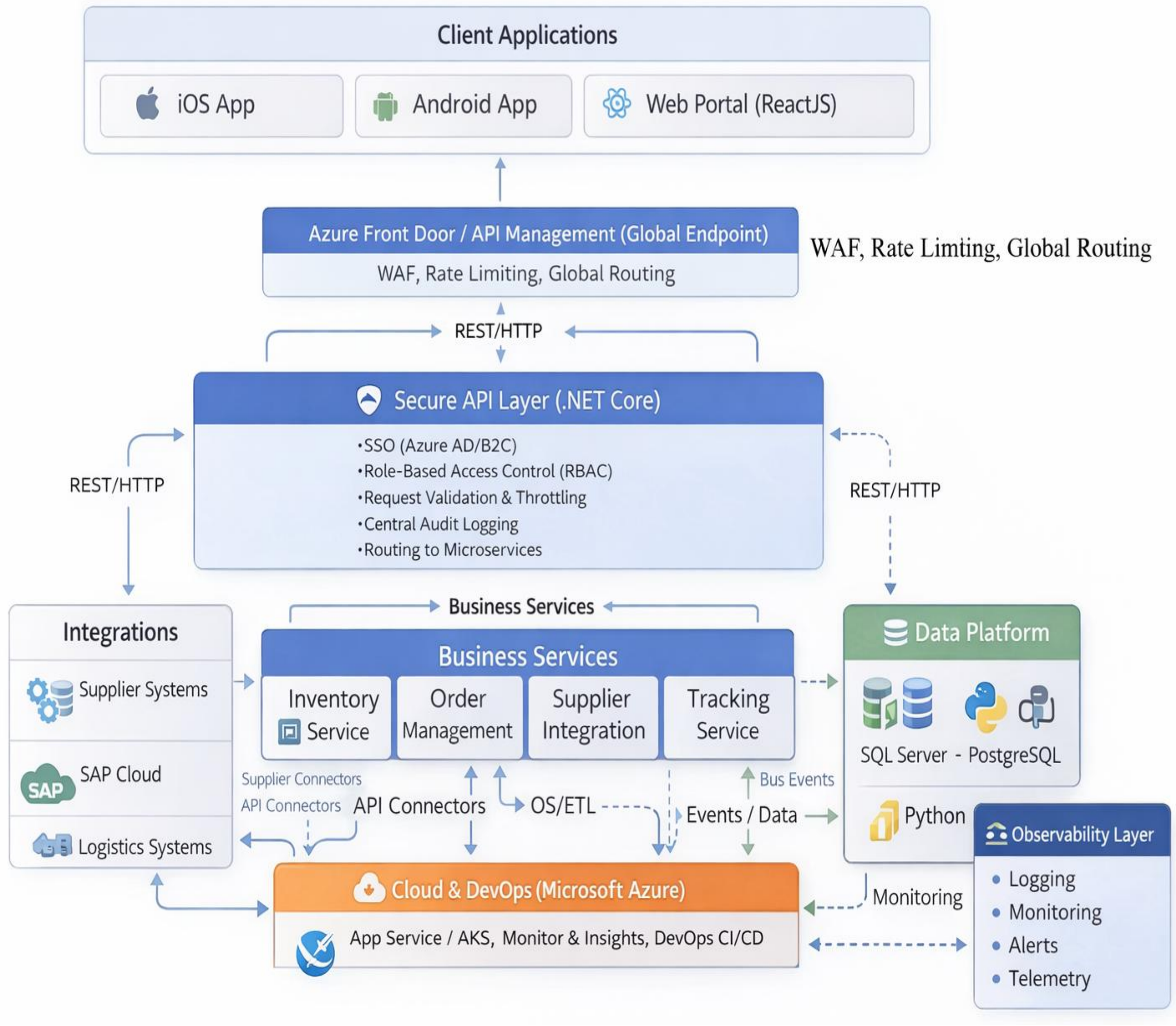
↓

Databases / Analytics

↓

Azure Infrastructure

REFERENCE ARCHITECTURE



Client Applications

iOS App

Android App

Web Portal (ReactJS)

Azure Front Door / API Management (Global Endpoint)

WAF, Rate Limiting, Global Routing

WAF, Rate Limting, Global Routing

REST/HTTP

Secure API Layer (.NET Core)

- •SSO (Azure AD/B2C)
- •Role-Based Access Control (RBAC)
- •Request Validation & Throttling
- •Central Audit Logging
- •Routing to Microservices

REST/HTTP

REST/HTTP

Business Services

Integrations

Supplier Systems

SAP Cloud

Logistics Systems

Business Services

Inventory Service

Order Management

Supplier Integration

Tracking Service

Data Platform

SQL Server - PostgreSQL

Python

Supplier Connectors

API Connectors

API Connectors

OS/ETL

Bus Events

Events / Data

Monitoring

Cloud & DevOps (Microsoft Azure)

App Service / AKS, Monitor & Insights, DevOps CI/CD

Observability Layer

- Logging
- Monitoring
- Alerts
- Telemetry

SCRUM RHYTHM

Artifacts

- ✓ Product backlog
- ✓ Sprint backlog
- ✓ Release roadmap

Key Delivery Metrics

- ✓ Sprint Velocity
- ✓ Sprint Burndown
- ✓ Defect Leakage
- ✓ Lead Time / Cycle Time

Sprint Planning
Define sprint goal and commit stories based on team capacity.

Daily Standups
15-minute sync to track progress, blockers, and dependencies.

Backlog Refinement
Mid-sprint grooming to prepare upcoming work items.

Sprint Review
Demo completed features to stakeholders for feedback.

Retrospective
Identify improvements to increase delivery efficiency.

DEFINITION OF READY & DONE

Definition of Ready

Story enters sprint only if:

- Business workflow identified
- NFR expectations defined
- API contract agreed
- External dependencies validated
- Data & monitoring impacts assessed

Definition of Done

Story completed only when:

- Feature validated (Web + Mobile)
- API performance benchmark met
- Security & RBAC verified
- Integration tests passed
- Monitoring dashboards updated
- No critical defects

Operational Controls

SLA Gate – performance validation

Observability Gate – structured logs & alerts

Data Integrity Gate – reporting consistency

Platform Consistency – Web & Mobile parity

👉 Backlog baseline established during Sprint-0 with scope changes managed through backlog reprioritization within fixed delivery capacity.

ENGINEERING EXCELLENCE & DELIVERY ACCELERATION

Day-0 Delivery Enablement

Architecture baseline finalized
Dev/QA environments & CI/CD ready
MVP backlog refined
Integration access validated

Toolchain

Azure DevOps

Git

PowerBI

Monitoring tools

Generative AI Engineering

AI assisted code generation

Automated test case generation

Code review and defect detection

Documentation generation

CI/CD Pipeline

Build →

Test →

Security scan →

Deploy

Reliability & Monitoring

API monitoring
Observability (Logging, Metrics, Alerts)

Test Pyramid

70% Unit

20% Integration

10% UI

PROJECT GOVERNANCE & RESOURCE LOADING

Delivery Team Structure

Role	# of Resources
Delivery Manager	1
Solution Architect	1
Scrum Master	2
Tech Leads	3
Backend Engineers	5
Frontend Engineers	4
Mobile Engineers	3
Business Analysts	2
QA Engineers	4
DevOps Engineers	2
Data Analyst	1
Total	**28**

(5 Delivery Pods comprising of backend, frontend, mobile, QA & DevOps)

RACI Matrix

Activity	Delivery Manager	Solution Architect	Product Owner	Scrum Master	Engineering
Architecture decisions	A	R	C	C	C
Backlog prioritization	C	C	R	C	I
Sprint planning	C	C	R	R	A
Delivery execution	A	C	C	R	R
Release approval	A	R	C	C	C

Legend:
R = Responsible
A = Accountable
C = Consulted
I = Informed

Governance Cadence

Daily

Scrum ceremonies

Scrum of Scrums

Weekly

Status report (WSR)

Sprint metrics review

Monthly

Client governance (MBR)

Delivery & financial health review

Release Checkpoints

Architecture review

Go-live readiness

MVP DELIVERY POD STRUCTURE

Pod 1 Web Platform (MVP Core)

Focus

Core supply chain workflows

Team

Tech Lead (1)

Backend Engineers (2)

Frontend Engineer (1)

QA Engineer (1)

Scope

Supplier onboarding

Inventory management

Order lifecycle

Core Platform APIs

Pod 2 – Web Platform (MVP Extensions)

Focus

Completing MVP user workflows

Team

Tech Lead (1)

Backend Engineers (1)

Frontend Engineer (1)

QA Engineer (1)

Scope

Shipment tracking

Supplier integrations

Web portal UX

Notifications

Pod 3 – Mobile Enablement

Focus

Mobile application enablement for supply chain workflows

Team

Tech Lead (1)

Mobile Engineers (2)

QA Engineer (1)

Scope

Shipment tracking

Supplier integrations

Web portal experience

Notifications

Pod 4 – Platform Engineering

Focus
External system integration and platform infrastructure enablement.

Team

Backend Engineer (1)

DevOps Engineers (2)

QA Engineer (1)

Scope

Supplier Integration APIs

SAP / logistics integrations

CI/CD pipelines

Infrastructure automation

Pod 5 – Data & Analytics

Focus
Operational analytics and data-driven supply chain insights.

Team

Data Analyst (1)

Frontend Engineer (1)

Mobile Engineer (1)

Backend Engineer (1)

Scope

Data pipelines

Reporting services

Power BI dashboards

Operational insights

MVP Delivery Pods : Scrum Master (1)

Platform Expansion Pods : Scrum Master (1)

Shared Program Governance: Delivery Manager (1), Solution Architect (1), Business Analysts (2)

👉 MVP delivered by end of Sprint 4

👉 Pods 1–2 focus on MVP delivery while Pods 3–5 build platform capabilities enabling rapid post-MVP expansion.

WORK BREAKDOWN STRUCTURE (WBS)

1. Discovery & Architecture

- Current platform assessment
- Target architecture definition
- Product backlog creation
- DevOps environment setup

2. Platform Foundation

- Authentication & RBAC
- Core API framework
- Web application framework

3. Integration Layer

- Supplier API integrations
- SAP cloud integration
- Logistics system integration

4. Supplier & User Onboarding

- Supplier registration workflows
- Catalog integration
- User role management

5. Inventory & Order Management

- Inventory tracking
- Order lifecycle workflows
- Shipment tracking

6. Mobile Applications

- Android application
- iOS application

7. Data & Analytics Platform

- Data pipelines
- Reporting services
- Power BI dashboards

8. Quality Engineering

- Unit testing
- Integration testing
- Automation framework

9. DevOps & Deployment

- CI/CD pipeline implementation
- Infrastructure automation
- Monitoring & observability

10. Production Readiness

- Performance testing
- Disaster recovery validation
- Security validation
- Production deployment

PROGRAM ESTIMATION & DELIVERY CAPACITY

Estimates

Workstream (Module)	Estimated SP
Discovery & Architecture	80
Platform Foundation	180
Integration Layer	220
Supplier & User Onboarding	160
Inventory & Order Management	220
Mobile Applications	200
Data & Analytics Platform	180
Quality Engineering	120
DevOps & Deployment	120
Production Readiness	80
Total Platform Size (in SPs)	**~1,600**

Delivery Capacity

Parameter	Value
Delivery Pods	5 pods
Pod Size	~5 engineers
Velocity per Pod	25–30 SP / sprint
Total Program Velocity	~150 SP / sprint
Sprint Duration	2 weeks

Timeline

Metric	Value
Total Estimated Size	~1600 SP
Program Velocity	~150 SP / sprint
Delivery Sprints	~10 sprints
Sprint 0	3 weeks
Stabilization & rollout	~4 weeks

ENGAGEMENT MODEL & COMMERCIAL STRUCTURE

Engagement Model Comparison

Model	**Managed Services (Recommended)**	**Time & Material**
Delivery Structure	Fixed delivery capacity (pods)	Flexible staffing
Cost Model	Fixed delivery envelope	Hourly billing
Delivery Scope	Defined delivery capacity (~1600 SP) with backlog prioritization	Dynamic backlog
Cost Predictability	High	Medium
Use Case	Core platform delivery	Continuous evolution

Managed Services (Fixed Capacity / Price)

Engagement Value: ~$1.2-1.35m

Target Margin: ~40–45%

Program Duration: ~6 months

Team Size: 25–30 engineers

Sprint Throughput: 150–170 SP / sprint

Total Delivery Capacity: ~1600 SP

Recommendation:
Managed services engagement provides predictable delivery capacity, stable pod structure and controlled cost while allowing backlog reprioritization across sprints.

RISKS & ASSUMPTIONS

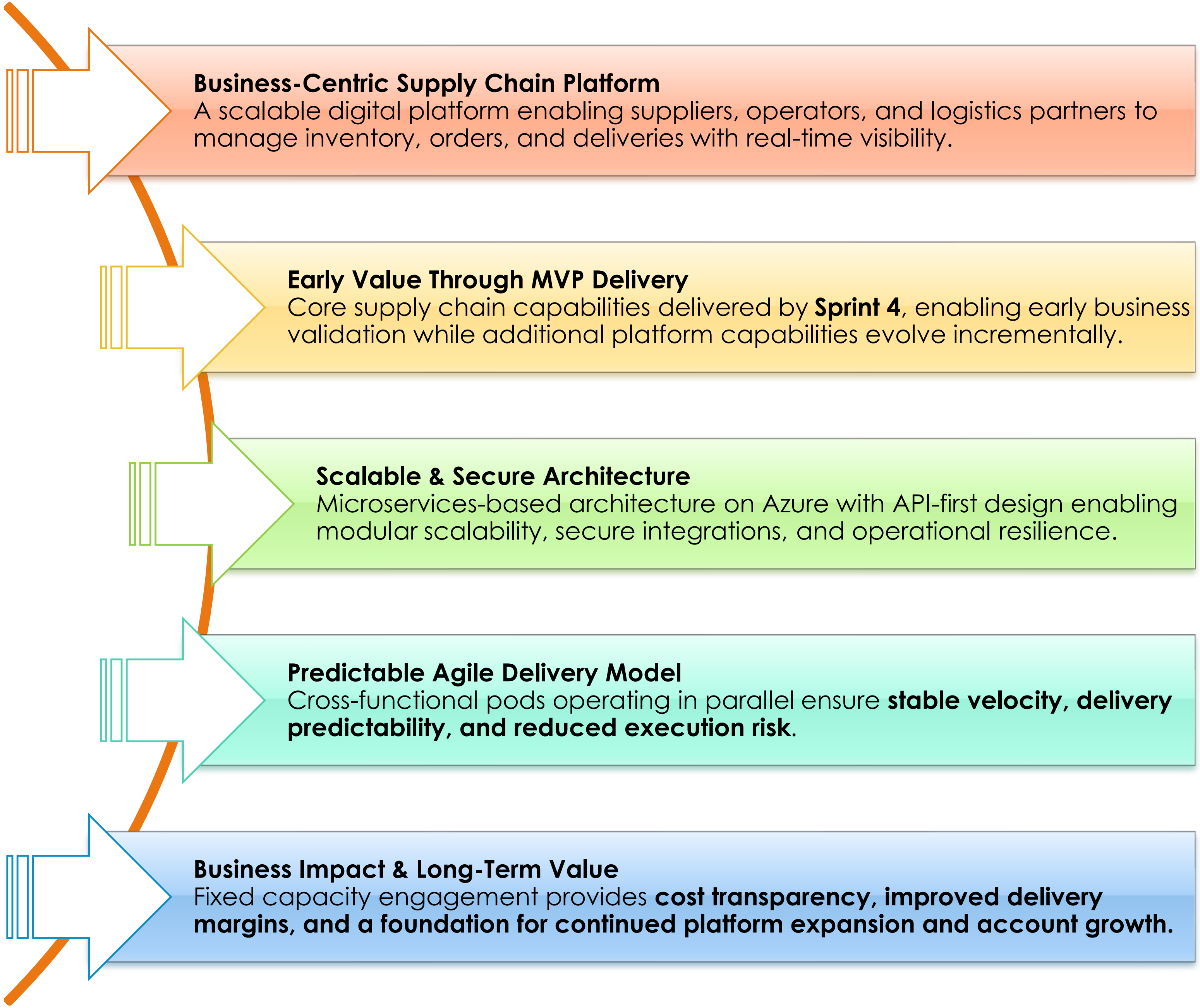
Risks

- Supplier integration complexity
- Data migration challenges
- Performance under peak demand
- Integration dependency risk
- Data quality inconsistencies

Assumptions

- Access to existing platform baseline and Azure infrastructure
- Client SMEs available for backlog refinement
- Supplier APIs accessible for integration
- Historical data available for analytics
- Business SLAs & KPIs clearly defined – for all types of clients

KEY TAKEAWAYS



Business-Centric Supply Chain Platform
A scalable digital platform enabling suppliers, operators, and logistics partners to manage inventory, orders, and deliveries with real-time visibility.

Early Value Through MVP Delivery
Core supply chain capabilities delivered by **Sprint 4**, enabling early business validation while additional platform capabilities evolve incrementally.

Scalable & Secure Architecture
Microservices-based architecture on Azure with API-first design enabling modular scalability, secure integrations, and operational resilience.

Predictable Agile Delivery Model
Cross-functional pods operating in parallel ensure **stable velocity, delivery predictability, and reduced execution risk.**

Business Impact & Long-Term Value
Fixed capacity engagement provides **cost transparency, improved delivery margins, and a foundation for continued platform expansion and account growth.**

THANK YOU!