Prodapt

Chase Extraordinary

Modernize to Move at Speed

A cloud-native order management can boost speed, scale, and operational efficiency

Credits | Srinivasan Murugesan | Senthil Velan | Sumit Thakur

Legacy order management IT stack hampers DSPs (Digital Service Providers) from achieving **business agility**, which is crucial to succeed in the digital world

Due to **heavy customization** and **multiple feature additions** done over the years, the legacy order management stack usually becomes a huge **monolithic application**

Major challenges DSPs face with the monolithic order management application

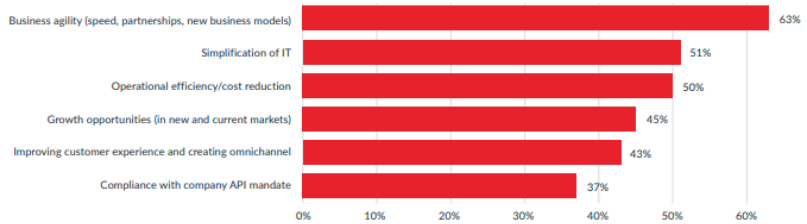
Long release cycles & complex dependencies. This hampers the time-to-market for new products.

Difficult to scale in delivering complex digital services to the customers**.**

Customer churn & loss of sales due to order fallouts and delays. New customers' first impressions are shaped by whether the provider can fulfill orders in a timely and accurate manner.

High OpEx to meet ongoing licensing, operations, and maintenance. This does not allow DSP's business to be competitive in the market.

But achieving business agility is crucial



TM Forum survey shows **business agility** as one of the top drivers for DSPs to initiate digital transformation

To overcome the above-stated challenges, there is a clear need for DSPs to **modernize their order management systems and processes.**

Chase Extraordinary

Prodapt

Typical approaches employed by DSPs for modernizing legacy order management stack

Leverage standard COTS products

Advantage

Lower initial cost and quick to implement.

Complexity

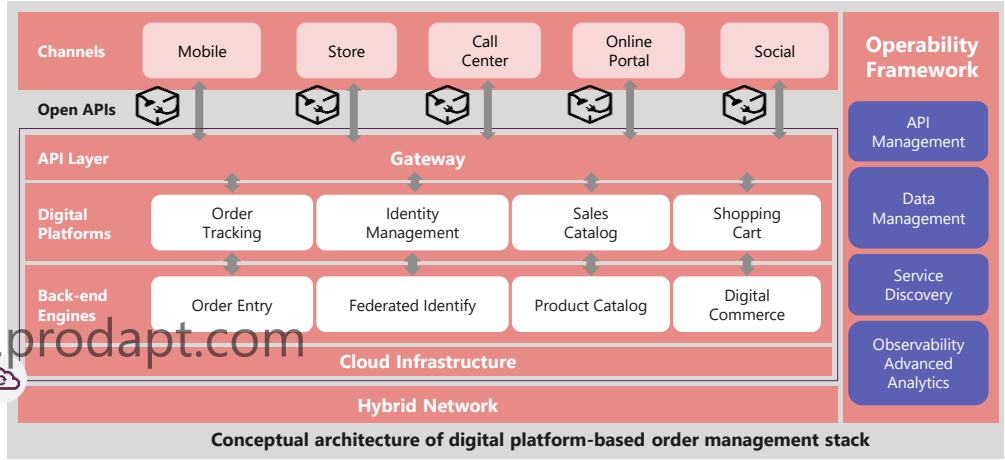
No one product in the market fits all business requirements.

- **Heavy customization done over the years makes the system no better than a monolithic application.** Customizations done initially might suit the business case, but year on year as new products are launched, the complexity increases and requires further customizations.
- **Selecting best-of-breed from different vendors often results in vendor lock-in.** Over time, this becomes an extremely fragmented and scattered approach.

Digital platform-based approach

A platform-based approach enables DSPs to create value by facilitating exchanges between two or more interdependent groups, usually consumers and 3rd party vendors.

Developing platform architecture requires breaking large monolithic applications into smaller, reusable components that can be accessed through **Open APIs**.



Conceptual architecture of digital platform-based order management stack

Advantage

- Enables DSPs to develop new business opportunities and compete faster in the digital economy.
- Once established, it's easier to create, build and manage complex services with improved customer-centricity.
- Improves business agility, provides smooth interoperability, and reduces costs.

Complexity

Establishing fully functional digital platforms is a complex program and if not strategized and implemented correctly can result in failure to meet the desired business outcomes.

RECOMMENDATION

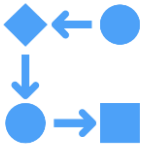
The *digital platform-based approach* promises to deliver long-term sustainable business benefits. Successfully implementing this approach requires the right set of transformation levers that are presented in the upcoming slides.

Key transformation levers to successfully modernize legacy order management stack by building a cloud-native digital platform

1

Business capabilities map
Identify core business capabilities and map to digital platforms

2



Architectural design guide
Ensure effective realization of the target state

3

Centralized product catalog
Reduce order fallout, enable zero-touch fulfillment and faster time-to-market of new products

4

Robust delivery pipeline
Provide continuous integration, deployment and monitoring of digital platform services

By embracing these transformation levers, DSPs can ensure a successful modernization of their order management stack, attaining a 50% reduction in time-to-market, ~30% OpEx reduction, and 40% increased productivity with E2E visibility in cloud-native applications.

Business capabilities map – Identify core business capabilities and map to digital platforms

This becomes the **foundation** of an overall order management transformation and is a vital step to compose required digital platforms.

This lever helps to segregate functionality and avoid interdependency during the development while ensuring the platform is more modular and agile.

RECOMMENDATIONS

Logically group the core order management capabilities

- Discover core order management capabilities and group them into domains defined by industry standards such as TM Forum (TAM, eTOM, ODA).

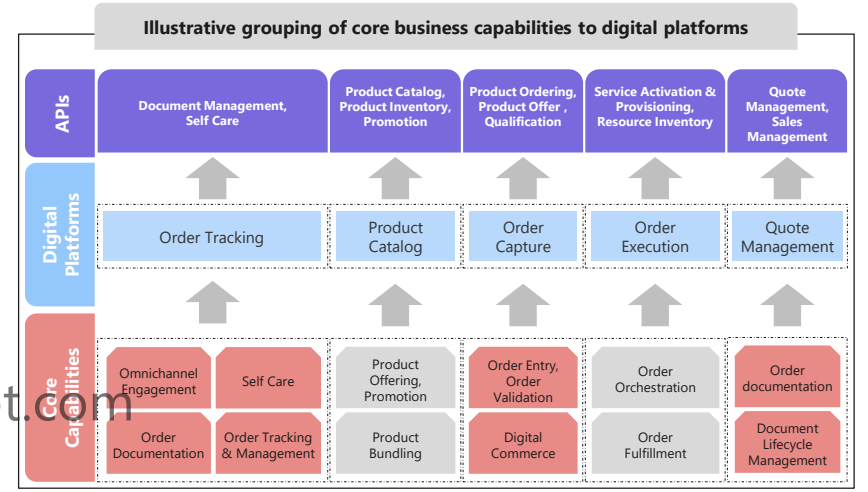
Define clear criteria for setting up digital platforms

- A platform should be big enough to provide an important and discrete service but small enough to be manageable. This should deliver a well-defined business capability (or a group of business capabilities).

Map identified core business capabilities to digital platforms and APIs

- Considering the digital platform criteria and the logical grouping of the core capabilities, map the digital platforms and relevant TMF Open APIs.
- E.g., order documentation, omnichannel engagement, self care, order tracking and management are all core capabilities that are mapped to the order tracking digital platform. Relevant TM Forum Open APIs here will be TMF910 - Document Management, TMF667 - Self Care API.

For DSPs, Order Management will typically comprise of 40+ core business capabilities mapped to 10 digital platforms with 60+ microservices and 120+ integration points.



Digital Platform Name	Description	Core Capabilities	Applicable TM Forum Open APIs
Order Capture	The Order Capture platform registers the orders from the customer independently of the sales channel used (internal distribution such as Telco Points of Sales, Outsourcer, external distribution such as Dealer, Reseller..., Call Centers, Customer Self-Service, or the channel medium used (face-to-face, phone, Web, email, mail/ fax...). It can be a new commercial offer order (for a new or an existing customer), the modification of the installed offer, products or the cancellation of some or all of the installed offers or products	Order Entry, Customer and Product Data Collection, Order Validation, Digital Commerce	Product Offering Qualification API, Product Ordering API, Shopping Cart API
Order Execution	Order Execution application enables enriches the missing parts of the order information, decomposes the Order and orchestrates the fulfillment of the order (Billing Order, Service Order, Shipment Order, Product Order etc)	Order Enrichment, Order Orchestration, Order Decomposition, Order Fulfillment	Service Ordering Management API, Service Activation and Configuration, Service Inventory Management, Resource Inventory Management, Resource Ordering Management,
Order Tracking	Functionality necessary to track and manage the distributed requests decomposed by Customer Order Distribution	Omni-channel Engagement, Self Care, Order Management,	Document Management API, Self
Lead Management	Lead Management applications provide functionality to man between prospective customers and account managers as w campaigns.		
Product Catalogue	Manage and synchronize product catalog across all channels hopping support, including managing the full process to buy or accesory. Provide recommendations like next best actions, provide the ability to compare plans/services, products based on features, price, availability, receive personalized offers		API, Recommendations API, Promotions API

Sample snippet of **business capabilities map** for order management

Architectural design guide for the effective realization of the target state

A successful platform should also integrate a variety of capabilities as well as the technologies & data behind them to deliver an integrated solution. This is crucial to create a well-structured platform and an excellent user experience.

RECOMMENDATIONS

Digital platforms should allow access to underlying data and business logic only via published APIs

- Legacy applications if any in the ecosystem should also be API enabled
- TM Forum Open APIs should be adopted wherever applicable and available

Choose the right API gateway which becomes a single-entry point from different channels/self-care applications to order management services

- **WSO2 API gateway** is recommended as it delivers an open-source platform-as-a-service for private and public clouds. With its componentized design, it provides seamless integration between servers, private clouds, and public clouds.

Use service mesh for order execution orchestration as this enables smooth communication between multiple microservices

- Avoid using API gateway for microservice-to-microservice communication. This creates too much overhead for a microservice as it ends up doing network communications, security authentication, handling timeouts and failures, load balancing, service discovery, etc.
- **Service mesh** is recommended in this scenario as it can offload all these network functions and handle the communication with all features.

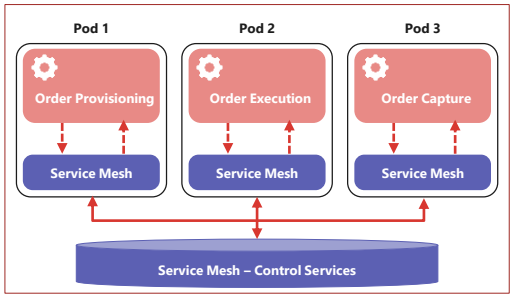


Fig: Using Service Mesh for order execution orchestration

Use Debezium to capture the change of order status

- Debezium provides a low latency data streaming platform for **change data capture**. On change of order status, this can trigger message to a notification service.

For order management, caching user session data is a key factor in building scalable and responsive applications

- Storing copies of the frequently used data on ephemeral but fast storage, improves application response time.
- **Write-through caching approach** is efficient as the cache sits between the application and the operational data store. The updates are done synchronously ensuring data consistency between cache and data store. **RedisGears** provides a write-through caching approach.

Architectural design guide for the effective realization of the target state

DSPs should embrace an **incremental approach** to transition to order management digital platforms. Transition architecture(s) should take care of consolidated gaps, dependencies matrix, and enterprise's capacity for creating and absorbing change.

RECOMMENDATIONS

Use strangler design pattern to enable incremental transformation

- Rewriting large monolithic order management from scratch involves huge effort and has a good amount of risk associated with it.
- Using strangler pattern reduces the above risk. Instead of rewriting the entire application, DSPs can incrementally replace particular functionality with a new service. This provides the business value of new functionalities much faster.
- Additionally, having new functionality integrated with an automated CI/CD pipeline makes it easier to deploy the microservices and can make the transition from monolith to microservices much smoother.

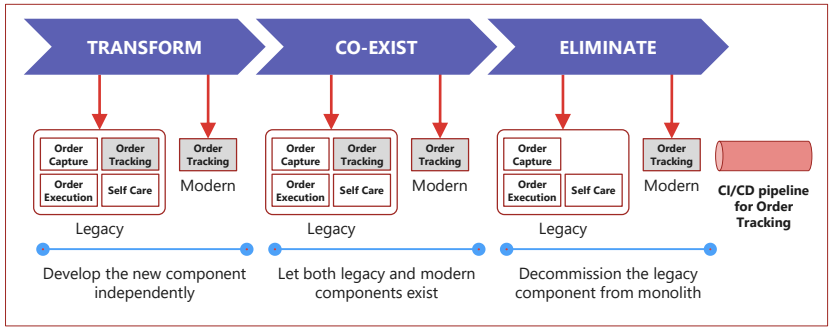


Fig: Illustration of order tracking service strangled from the monolith into an independently deployable service

Implement an anti-corruption layer to maintain access between new and legacy subsystems without corrupting/compromising the design of the new system

- Legacy systems often suffer from quality issues such as obsolete APIs or convoluted data schemas.
- Placing an anti-corruption layer in between helps to translate the communications, while allowing the new system to remain unchanged and can avoid compromising its design approach.
- The layer can be retired after all legacy functionality has been migrated.

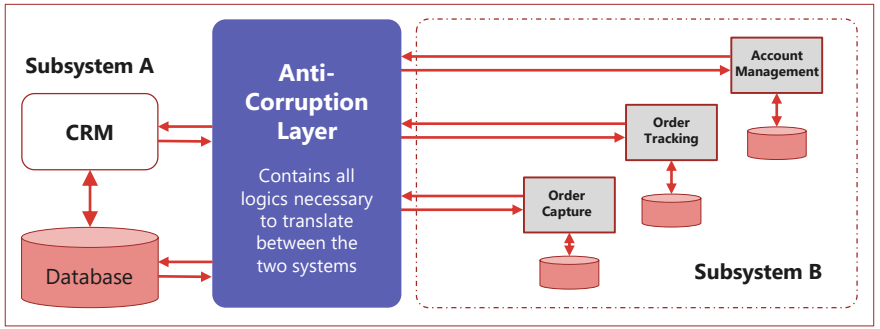


Fig: Illustration of anti-corruption layer to maintain access between subsystems

Centralized product catalog – Reduce order fallout, enable zero-touch fulfillment and faster time-to-market for new products

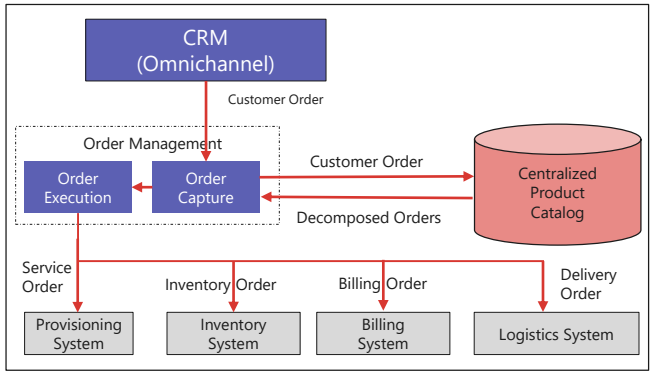


Fig: Centralized Product Catalog

BENEFITS

- **End-to-end visualization** enables rapid modeling of new products, change existing products and validate the end-to-end configurations at design time.
- **Inflight changes** allow product and marketing teams to adapt their offerings quickly, while delivery and operations teams can extend configurations for fulfillment using the same master model.
- **Overcomes fulfillment challenges** by leveraging data centralized in the catalog to drive ordering flows. This reduces dependency on local reference data in order management systems.

RECOMMENDATIONS

Apply **data federation** by implementing a **new central product catalog** that can be synchronized with existing product catalogs and enable one integration point for all O/BSS solutions.

Generate dynamic orchestration plan comprising of:

- Decomposition and enrichment of basic products into Customer Facing Service (CFS) and Resource Facing Service (RFS)
- Associated actions on products such as shipping, provisioning, and billing
- Sequencing of products

Offers	Broadband 599 plan (300 Mbps, unlimited data)		Mobile 299 plan (5G, 10GB data, unlimited calls)		
Products	Broadband bandwidth	Broadband data	Mobile Network (4g/5g)	Mobile data	Mobile call usage
Customer Facing Service (CFS)	Broadband internet access	Broadband data	Mobile Tariff XXX		
Resource Facing Service (CFS)	ADSL	Fibre	MSISDN	ICCID	
Resources	Router	LAN/Ethernet	SIM	E-SIM	
Actions	Shipping	Provisioning	Billing		

Fig: Illustration of dynamic orchestration plan

Achieve interoperability and interfacing by adopting the TM Forum SID model

- Identification of Open APIs relevant to digital platform becomes easier by adopting the data model and REST specification provided by TM Forum along with the centralized catalog. E.g., TM Forum Product Catalog Management API (TMF620), Product Inventory Management API (TMF637) can be used here.

Robust delivery pipeline - Provide continuous integration, deployment and monitoring of digital platform services

Leverage elastic hosting infrastructure with container orchestration application like OpenShift

- Setup **one-click new environment on OpenShift.** This should provide pipeline aggregation, DevOps analytics, and release management in one click.

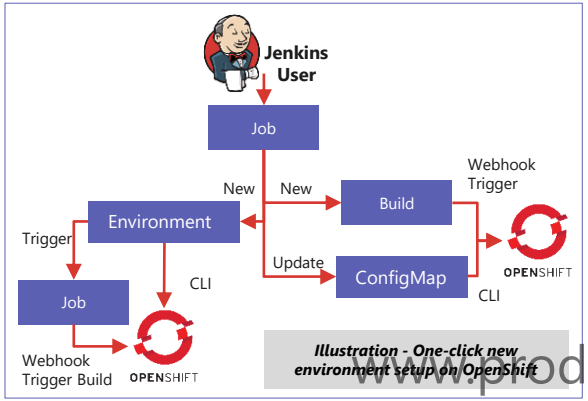


Illustration - One-click new environment setup on OpenShift

Automate database schema changes into DevOps process

- An open-source solution such as **Liquibase** addresses problems associated with manually updating database schemas. It automates tracking, managing, and applying database schema changes to ensure faster software releases.

Take a proactive approach in order fallouts management

- Enhance the error handler to stamp the correlation ID to the fault message and publish it to the error topic. This helps to uniquely identify an order fallout fault message.
- Provide additional information to capture pertinent data about the order failure.
- Ensure the fallout management integration process contains detection, notification, and correction mechanisms.

Use observability techniques to effectively monitor order management applications

- Implementing observability techniques by collectively analyzing logs, traces, and metrices can be a lot more beneficial in the complex cloud-native environment.
- This requires DSPs to efficiently implement observability techniques to gain critical insights.

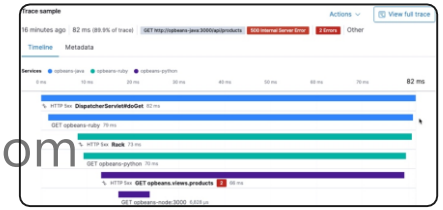


Fig: Observing the request flow using trace samples

Increases productivity by 40% with better workflows for debugging and performance optimization

Build a central governance platform

- Digital platform built with microservice approach can result in creating numerous services implemented by autonomous teams with different SLAs, technologies, message formats, etc.
- Building a central governance platform to track all the details about these services will be key when the platform expands with more new services.
- **Key features to be considered in governance layer** are discovery, search, documentation, Lifecycle Management (LCM), and comments/reviews/forums.

Benefits achieved by a leading DSP in Europe after transforming their legacy order management to cloud-native digital platform

Implementing the key transformation levers as discussed in this insight, resulted in the following benefits.

Improved business agility
- 50% reduction in time-to-market
- 30% reduction in OpEx
- 40% increased productivity with E2E visibility in cloud-native applications

Production rollout time reduced by 75%, thereby accelerating the overall time-to-market.

New environment setup time reduced by 66%

Cloud scalable containerization for integrations

Adherence to TM Forum Open API standards

Improved customer experience due to faster order fulfillment

Get in touch

USA

Prodapt North America, Inc.
Oregon: 10260 SW Greenburg Road, Portland
Phone: +1 503 636 3737

Dallas: 1333, Corporate Dr., Suite 101, Irving
Phone: +1 972 201 9009

New York: 1 Bridge Street, Irvington
Phone: +1 646 403 8161

CANADA

Prodapt Canada, Inc.
Vancouver: 777, Hornby Street,
Suite 600, BC V6Z 1S4
Phone: +1 503 210 0107

PANAMA

Prodapt Panama, Inc.
Panama Pacifico: Suite No 206, Building 3815
Phone: +1 503 636 3737

CHILE

Prodapt Chile SPA
Las Condes: Avenida Amperico Vespucio Sur 100, 11th Floor, Santiago de Chile

UK

Prodapt (UK) Limited
Reading: Suite 277, 200 Brook Drive,
Green Park, RG2 6UB
Phone: +44 (0) 11 8900 1068

IRELAND

Prodapt Ireland Limited
Dublin: Suite 3, One earlsfort centre,
lower hatch street
Phone: +44 (0) 11 8900 1068

EUROPE

Prodapt Solutions Europe &
Prodapt Consulting B.V.
Rijswijk: De Bruyn Kopsstraat 14
Phone: +31 (0) 70 4140722

Prodapt Germany GmbH
Münich: Brienner Straße 12, 80333
Phone: +31 (0) 70 4140722

Prodapt Digital Solution LLC
Zagreb: Grand Centar,
Hektorovićeva ulica 2, 10 000

Prodapt Switzerland GmbH
Zurich: Muhlebachstrasse 54,
8008 Zürich

Prodapt Austria GmbH
Vienna: Karlsplatz 3/19 1010
Phone: +31 (0) 70 4140722

Prodapt Slovakia j.s.a
Bratislava: Plynárenská 7/A, 821 09

SOUTH AFRICA

Prodapt SA (Pty) Ltd.
Johannesburg: No. 3, 3rd Avenue, Rivonia
Phone: +27 (0) 11 259 4000

INDIA

Prodapt Solutions Pvt. Ltd.
Chennai: Prince Infocity II, OMR
Phone: +91 44 4903 3000

"Chennai One" SEZ, Thoraipakkam
Phone: +91 44 4230 2300

IIT Madras Research Park II,
3rd floor, Kanagam Road, Taramani
Phone: +91 44 4903 3020

Bangalore: "CareerNet Campus"
2nd floor, No. 53, Devarabisana Halli,
Phone: +91 80 4655 7008

Hyderabad: Workafella Cyber Crown 4th Floor,
Sec II Village, HUDA Techno, Madhapur

THANK YOU!