# Enterprise Connected Services

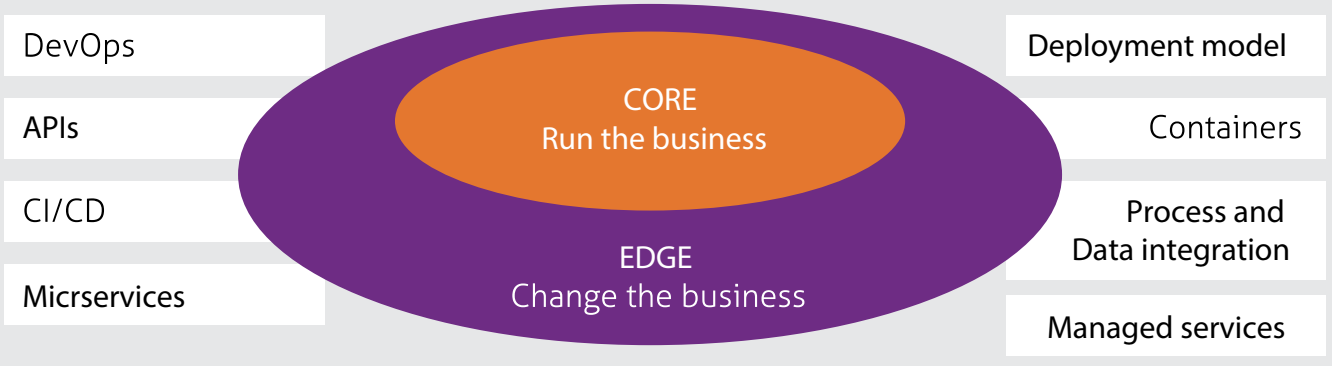
As digital engagements and adoptions are exponentially increasing among consumers across industries and channels, entire technology ecosystem (including vendors, customers, partners, system integrators) is under pressure to improve efficiency, provide best lifestyle support, increase profitability, and transform their value proposition in order to thrive and sustain business models in today's hyper connected world.

Having a viable digital strategy can make the difference between transforming ecosystem's business and capitalizing on opportunities based on adoption of emerging technologies. Application Programming Interfaces (APIs) play a foundational role in realizing this strategy.

LTIMindtree brings in technology expertise in order to respond to increasingly connected consumers with intelligent, personalized, integrated, and delightful experiences. We recognize the trends-together customer requirements and realize the strategy in order to capitalize on the big shift. Our continuous mantra has been 'API First' throughout Enterprise Connected Services (ECS) integration practice.

It opens door for agile innovation, quickly adapt to market changes, new competitors, and capitalize the opportunity to gather wealth of data about products, services, apps, developers, and users as single ecosystem. Enables business to participate in the digital economy.

## Integration practice complete enablement and journey for enterprise



Integration practice leverages experiences from different projects and stakeholders of an enterprise to react quickly to the dynamic nature of user requirements and business execution. Mission critical core processes (also known as "core services") are still managed and operated by central IT with system integrators and vendors as managed services. These services run the business and change rather frequently.

Other side of the spectrum is the line-of-business trying out new or adapt to business/digital disruptions quickly in an agile way. Innovation by 'fail-fast' strategy is experienced and practiced, which in turn is creating "edge services". Edge services is getting more and more importance to enhance or disrupt existing business models and therefore changing the business.

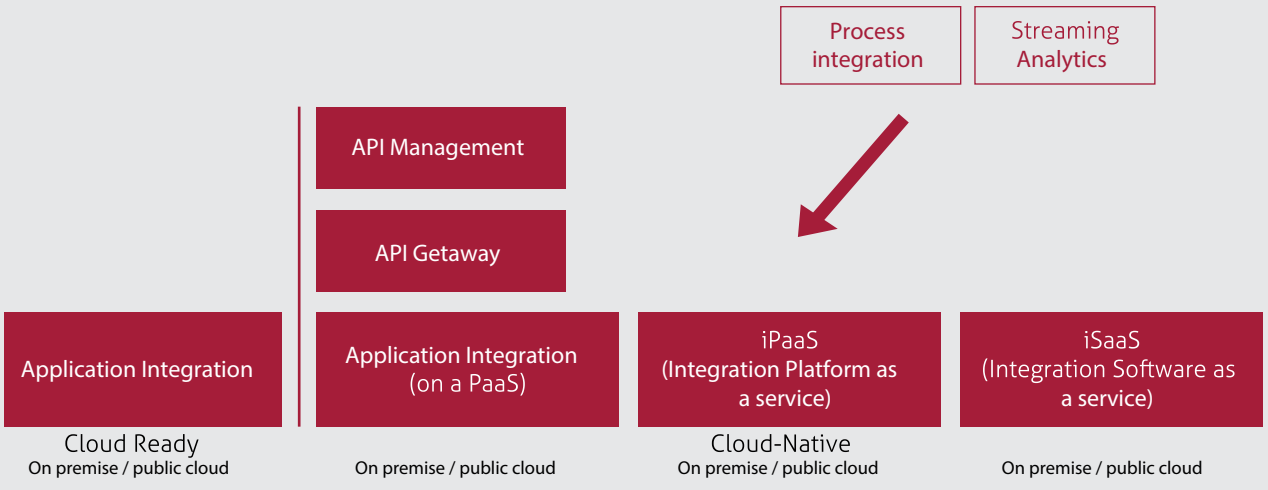
LTIMindtree's Integration Strategy is purely driven by our commitment and continuous engagements with customers. Customer engagements have been primarily focused into consultancy, architecture assessment, implementations, delivery, support, partner collaboration, presales support, and operations for HIP (Hybrid Integration Platform) enablement.

## API Predictions and Trends

- API Consumption at a rapid rate
- API Discovery and Marketplaces are springing up to facilitate API consumption
- API Product Management and Full Lifecycle API Management will see more increase in adoption
- Graph QL and Nodejs will continue to grow quickly
- Open API specification will gain wider adoption
- Bots, AI, and IoT will continue to drive API adoption
- Microservices , DevOps will become more closely associated with API Gateway (Integrate, Secure, and Monetize Microservices)
- Server-less architecture will be adopted by all cloud /HIP (Hybrid Integration Platform) providers
- IoT Security mandatory clause will force API security

LTIMindtree's well established business working model and technical frameworks or accelerators around Continuous Delivery, focus areas have been into – Microservices, Process Integration, Data Integration, EDA (Event Driven Architecture), API Management and Lifecycle adoption, iSaaS iPaaS based solution as well as deployment, Streaming Analytics, IoT Integration, Big Data, Data Lake and Big Memory Integration and Adoption, Cognitive Computing (including AI, Machine Learning), DevOps adoption from middleware perspectives, and BPM driven IT transformation. Focus areas have been evolved to align with Hybrid Integration Platform (HIP).

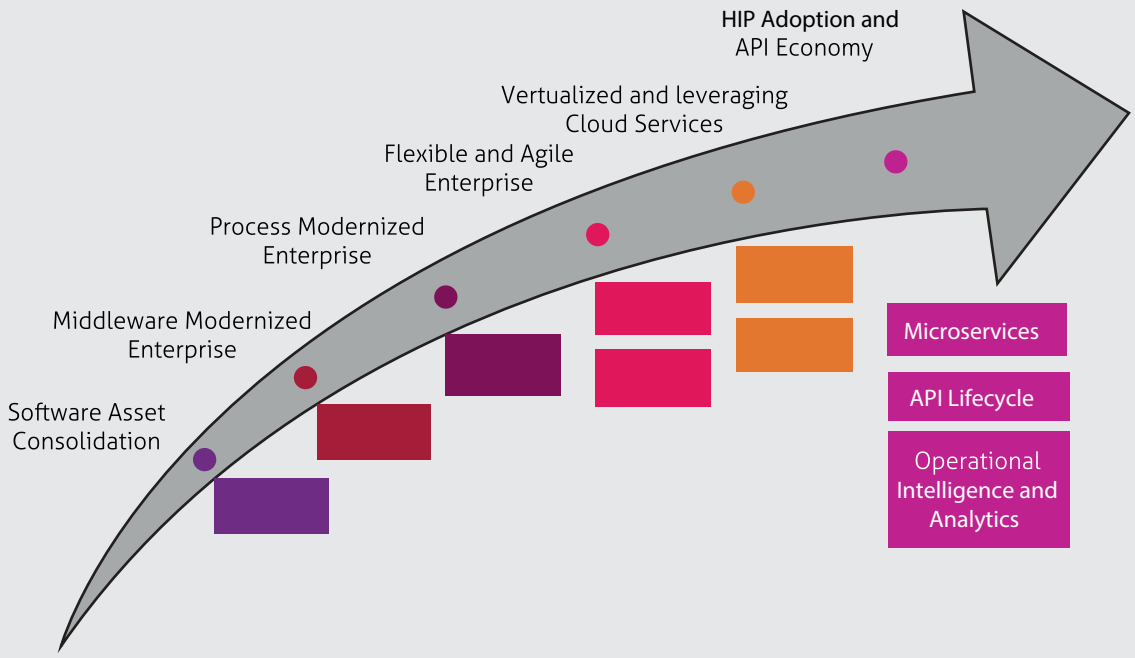
Components required for HIP to run and change the business successfully is shown as:




### Benefits reaped through LTIMindtree's Enterprise Connected Services (ECS) are described as:

- Independently deployable and scalable services (microservices based approach) in addition to existing EAI and SOA driven implementations. Starts with enabling customer through technical workshops and sessions as awareness into microservices architectural style
- Domain driven design pattern for projects like Data Lake and common integration platform driven by APIs is experienced where business domains/LOBs enforce segregation of monoliths and organized around business capabilities. De-centralized governance and data management brings in more operational efficiency from infrastructure, production support, operations, and DevOps support structure teams
- DevOps first approach is currently practiced by customers where there is joint collaboration between SI, developers/architects community within customers and System

## LTIMindtree's HIP evolution and journey

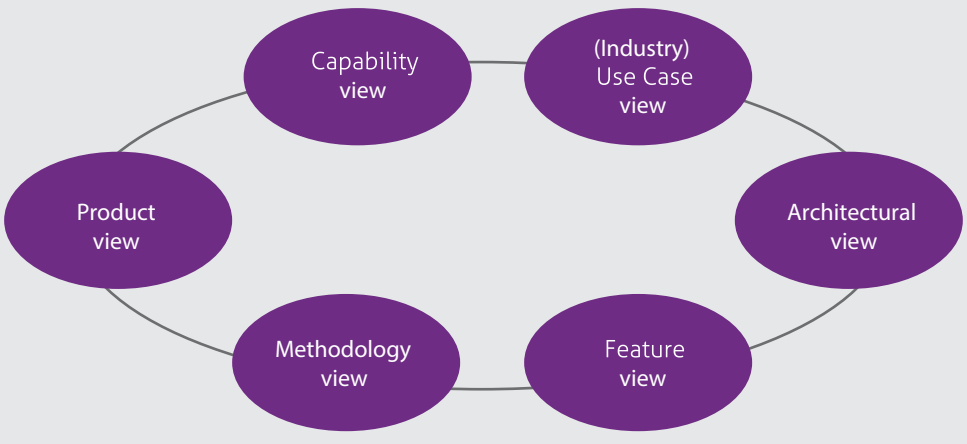



Integrators and operations team. For projects involving initial architecture with solution, physical, and logical –stress have been mainly into – a. deployment models (either on-premise/hybrid/cloud based), b. architectural guide lines, policies and principles, sizing and capacity planning (Elasticity without downtime) based on volumetric analysis and forecasting of TPS, c. Design and Configuration (Automated Configuration management, Continuous Release aligned with agile practice in multiple defined sprints, application build, and deployment process), d. Automation during implementation phase achieved through continuous integration, continuous monitoring, installation and setup, continuous testing, e. Operational intelligence – business services profiling, performance optimization, KPIs monitoring (business as well as system specific to supporting infrastructure)

- Responsive and Resiliency – With adoption of API, CIP, Mediation, ESB, and BPM stack along the core hybrid integration platform continuously measuring application response and providing resilient mechanisms have become an utmost priority starting with complying with mandatory compliances. Ensuring resiliency with processes, standards, functions, and entire value chain drives complete solution to meet NFR requirements through out CoE practices

A holistic view of our ECS consists of Product view, Capability view, Industry Use Case view, Architectural view, Feature, and Methodology view. These view are taken into consideration during consultancy, architectural assessment, technical proposal, presales/post sales and execution/ delivery engagements.

## Digital Integration Platform



## Product View

Practice team provides competitive intelligence and analysis as part of product view to customers during evaluation and assessment phases of middleware/API products.

| Product Features | Description |
|---|---|
| Enterprise Architecture (EA) | Provides architecture guidelines, principles, and design patterns. Practice focusses on artifacts associated with EAI, SOA, BPM, API engagements |
| Enterprise Service Bus (ESB) | ESB forms the core backbone for middleware practice and adoption within customer's projects |
| Model to Execute (M2E), BPM | Understanding Business Process Management (BPM) as a management practice Includes both organizational and technical implementation projects.<br>Uncovers following challenges:<br>■ How will business requirements be implemented?<br>■ How will business and IT work in concert?<br>■ How will work progress and be synchronized between business and IT?<br>■ How will transparency be assured during implementation?<br>■ How will change request for already implemented processes be handled?<br>To meet these challenges, M2E is proposed, while runtime execution of M2E requires BPM stack? |
| Operational Intelligence and Streaming Analytics | Streaming analytics, also called event stream processing, is the analysis of large, in-motion data called event streams. These streams comprise events that occur as the result of an action or set of actions, such as a financial transaction, equipment failure, or some other trigger.<br><br>Operational Intelligence (OI) is a form of real-time dynamic, business analytics that delivers visibility and insight into business operations. OI solutions run query analysis against live feeds and event data to deliver real-time visibility and insight into business and IT operations. |
| Messaging | An enterprise messaging system (EMS) is a set of published enterprise-wide standards that allows organizations to send semantically precise messages between computer systems. EMS systems promote loosely coupled architectures that allow changes in the formats of messages to have minimum impact on message subscribers. |
| In Memory Data Fabric (IMDF) | Comprehensive in-memory data platform that includes data grid, compute grid, complex event processing, and real-time streaming |
| DevOps (Middleware context) | DevOps practice/scenarios for middleware reflects different usage patterns for middleware application development, improvement, and maintenance. DevOps in this context provides guidance to – a. Accelerate software delivery, b. Balance speed, cost, quality and risk, c. Reduce time to feedback |
| Microservices | Microservices is a specialization of an implementation approach for service-oriented architectures (SOA) used to build flexible, independently deployable software systems. In a microservices architecture, services should have a small granularity and the protocols should be lightweight. |
| Deployment Model | There are different deployment models required for applications to be running. We focus, on three kinds of deployment models and practices – On-Premise, Public Cloud, and Hybrid models. Based on different cloud computing practices like iSaaS and iPaaS, deployment models are proposed accordingly. |

| | |
|---|---|
| Application to Application (A2A) Integration with MDM | Master Data management is practiced from application to application integration for federated as well as centralized data hubs. In digital integration practice, some of the design patterns and frameworks are developed to facilitate data lake initiatives and data driven workflows. |
| Governance | Governance adopted in HIP and iPaaS scenarios, practice could easily manage service interfaces as well as ensures all the system interactions are secure and conform to policies and standards as applicable. |

## Capability View

Capability view is formulated by ECS practice team based on technology categories and their support to integration scenarios, patterns, deployment and operational models and governance.

| | Integration Scenario | | | | Deployment Model | | | Operation Model | | Governance | | | | Integration Patterns | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ECS Practice Areas | A2A | B2B | CSI | IoT Integration | On Premise | Cloud | Hybrid | ECS Practice Managed | Hybrid | Policy Managed | Policy Enforcement | Policy Enforcement | Service / API Catalogue | Composite Application | Orchestration | Service / API Façade | Data Factory / Integration | CAP |
| API Management | | | | | | | | | | | | | | | | | | |
| On Premises | | | | | | | | | | | | | | | | | | |
| B2B Integration | | | | | | | | | | | | | | | | | | |
| Data Integration | | | | | | | | | | | | | | | | | | |
| Integration Frameworks | | | | | | | | | | | | | | | | | | |
| Process Integration | | | | | | | | | | | | | | | | | | |
| iPaaS | | | | | | | | | | | | | | | | | | |
| iSaaS | | | | | | | | | | | | | | | | | | |
| IoT Platform Integration | | | | | | | | | | | | | | | | | | |
| Streaming Analytics | | | | | | | | | | | | | | | | | | |
| In Memory Data Fabric | | | | | | | | | | | | | | | | | | |
| Cognitive Computing | | | | | | | | | | | | | | | | | | |
| DevOps | | | | | | | | | | | | | | | | | | |
| Microservices | | | | | | | | | | | | | | | | | | |

Enterprise Connected Services (ECS) Practice and Delivery

A2A – Application to Application Integration
CSI – Consulting and Solution Implementation

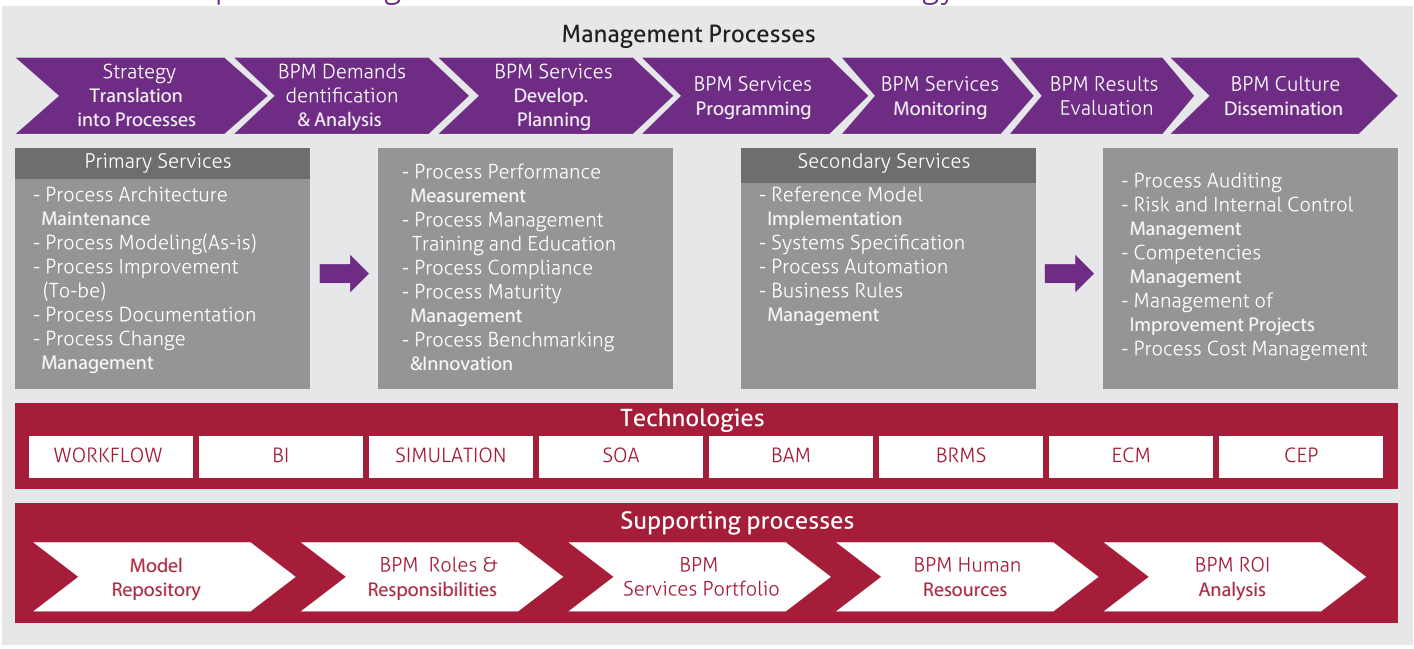
CAP – Consistency, Availability and Partition Tolerance
B2B – Business to Business

## Methodology View

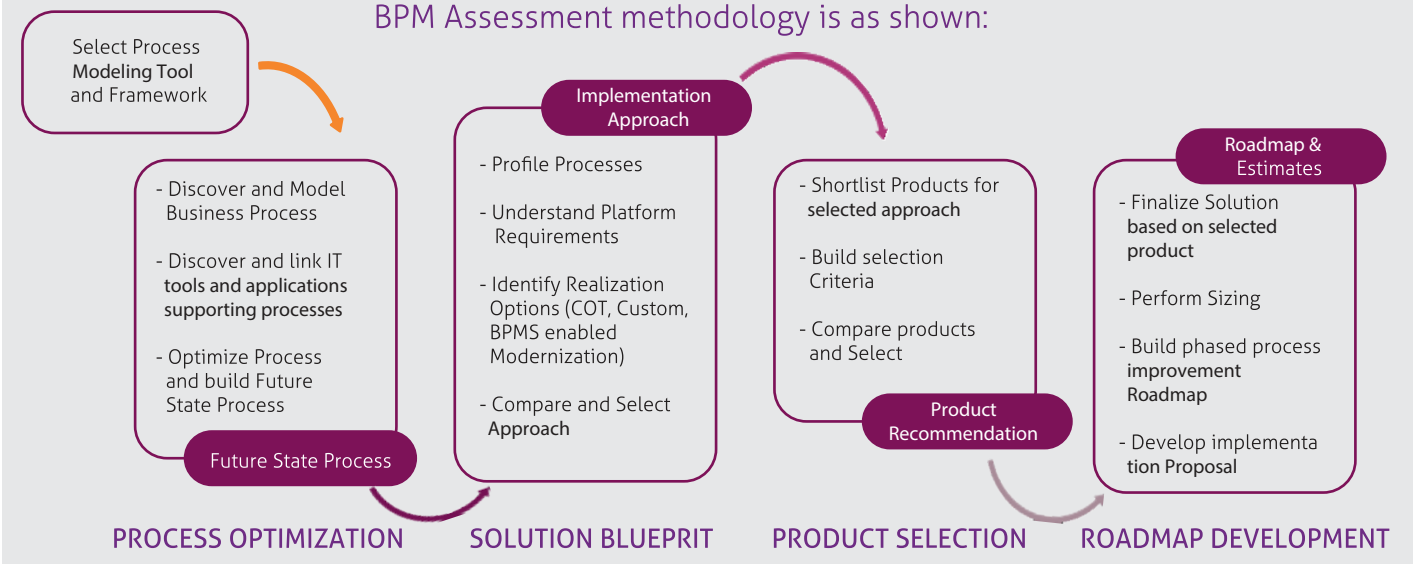
ECS follow CoE adopted methodology for API, Integration, and BPM based solutions and projects. Also includes frameworks, accelerators in the form of utilities or tools and artifacts associated with execution model and automated testing. Methodology activities or tasks are classified as assessment, implementation, and operations.

| Assessment | Implementation | Operations |
|---|---|---|
| BPM Architecture | BPM project execution | Configuration Management |
| BPM DevOps driven | BPM Transformation model | DevOps Cycle |
| BPM Runtime Intelligence | EAI/SOA Transformation | Microservices based transformation |
| EAI Architecture | Integration and UAT Testing | iPaaS deployment |
| SOA Architecture | API Lifecycle Management | API environment administration |
| Microservices Architecture | | API monitoring |
| API Architecture, API adoption strategy | | |

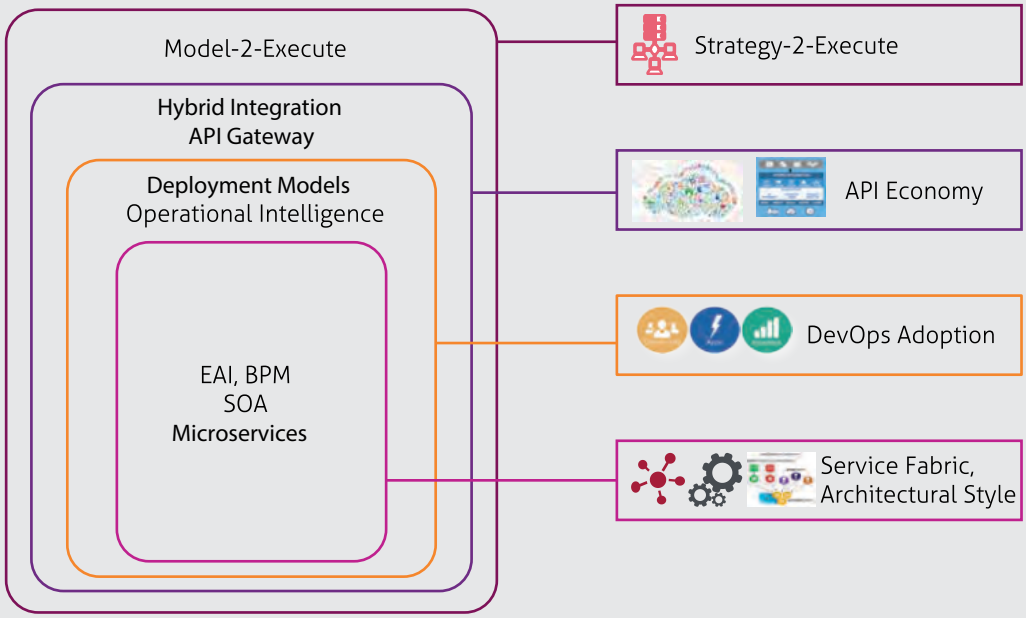
## Example including BPM CoE framework and methodology is included as shown:




## BPM Assessment methodology is as shown:



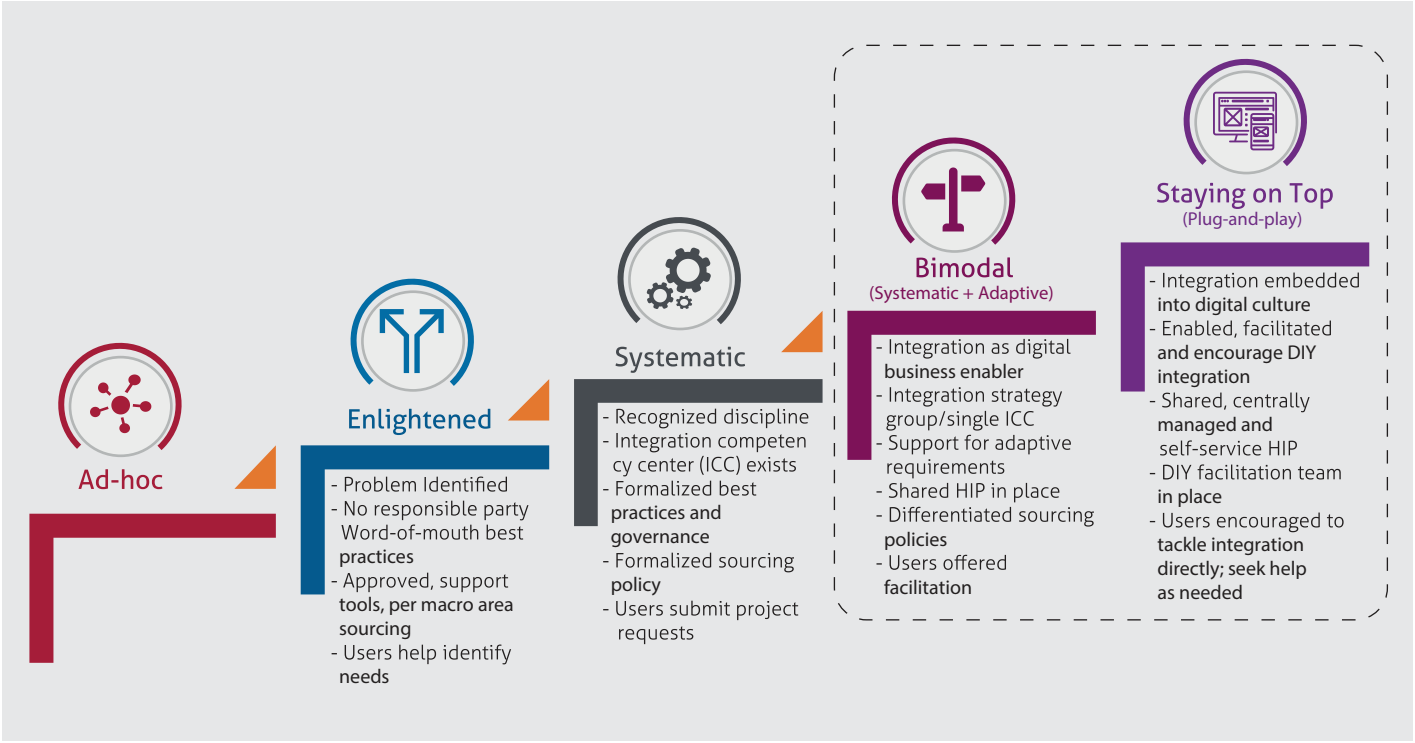

ECS practice focusses on adopting methodologies based on LTIMindtree's extensive experience into middleware integration practice.



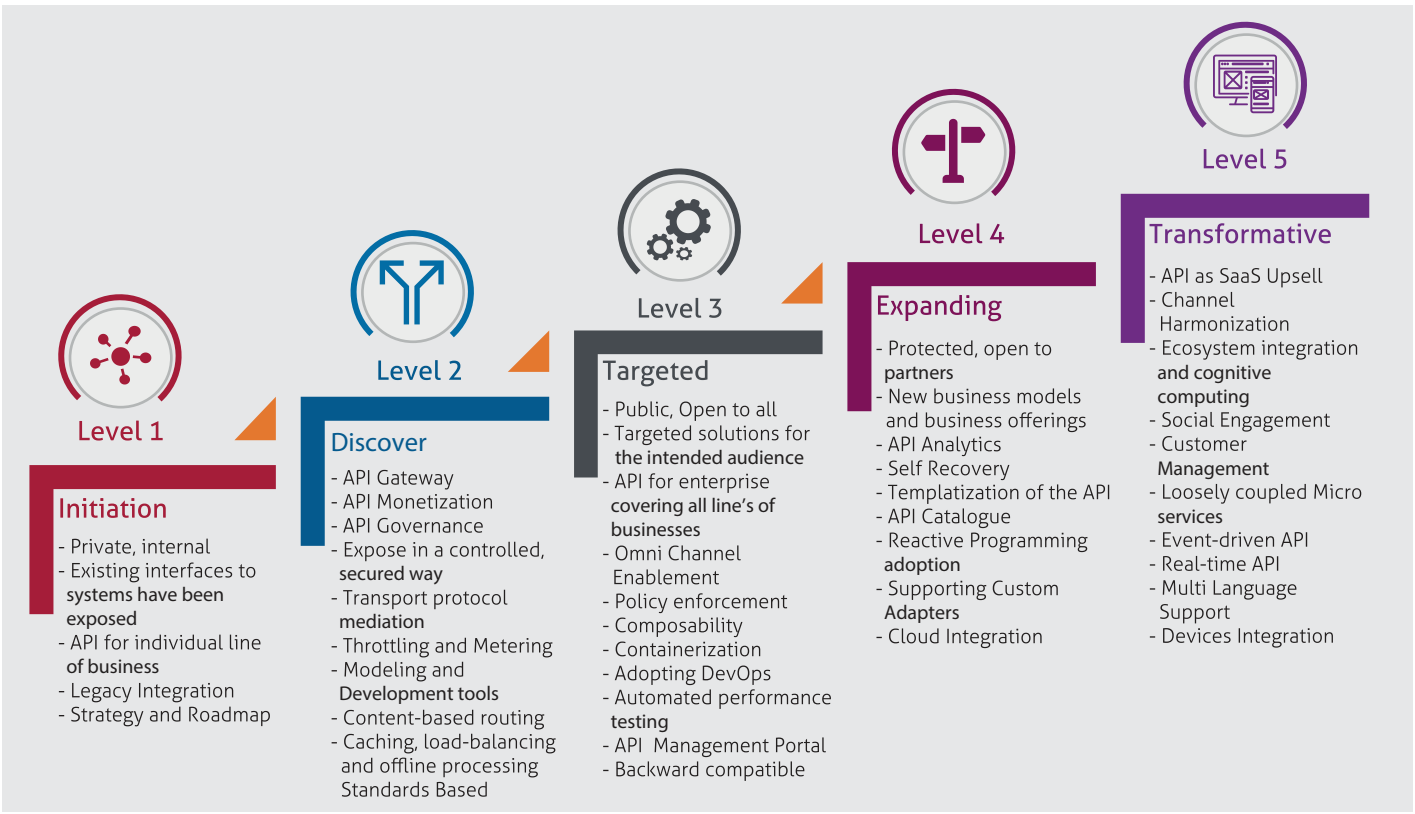
LTIMindtree ECS practice provides initial assessment of the integration adoption and maturity along with vertical practice engagements. Depending on requirements, consultants with various skills and level of experience are engaged into core integration projects.

## ECS Practice and API Maturity Model

LTIMindtree's ECS Practice maturity model is focused on either Bimodal (Systematic + Adaptive) or Staying-on-Top (Plug-and-Play)




Each of these aspects are religiously reviewed with assessment at customer's API adoption engagements. With evolving nature of digital business platform, integration practice also keeps evolving with new evaluation and maturity assessment criterion.

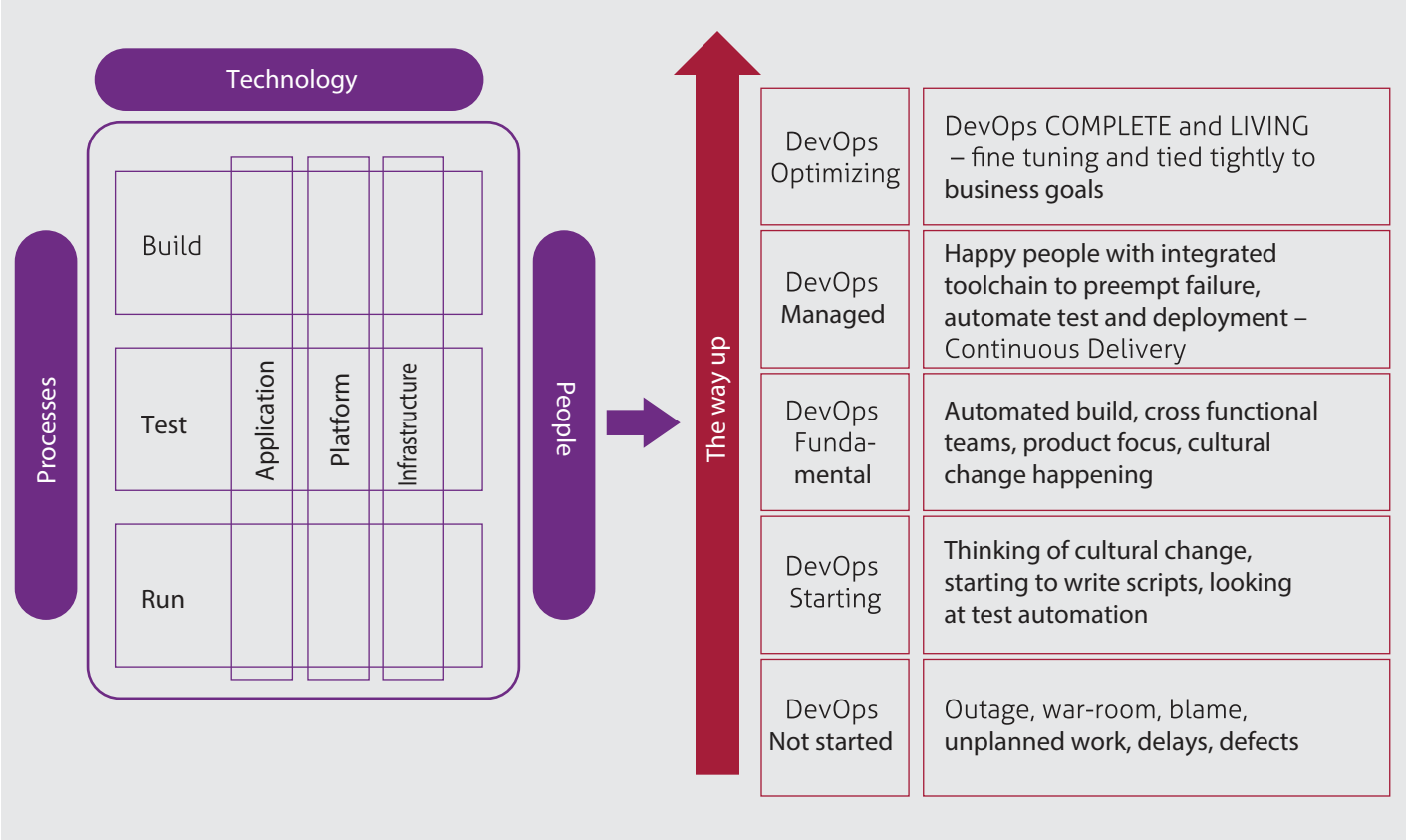



API Maturity is focused on practicing and adoption of microservices based approach, SOA based implementations covering a journey from Megalith platform to Monolith platform to Macro SOA platform to Meso application platform to Microservices platform.

## DevOps Adoption with ECS practices and delivery

API maturity is also assessed through adoption of DevOps with the following aspects of practice and tools deployed during phases:

- Application performance management
- Monitoring and health check
- End user experience monitoring
- Analytics and operational intelligence
- Incident management
- Continuous integration
- Continuous build automation
- Automated configuration management
- Continuous testing
- Continuous delivery and release
- Artifact repository
- REST API
- Source code management
- Release automation
- Containers
- Microservices
- QA automation
- Load testing
- Performance testing
- Log analytics
- On-Demand application environment
- Collaboration and communication
- DevOps integration
- Adaptive security
- Backup and recovery




In addition, KPI driven DevOps metrics is also monitored and adopted through the engagements.

- Deployment frequency
- Change volume
- Lead time (From development to deployment)
- Percentage of failed deployments
- Mean time to recovery (MTTR)
- Customer ticket volume
- % change in user volume
- Availability
- Performance (Response time)

## API Strategy and Landscape

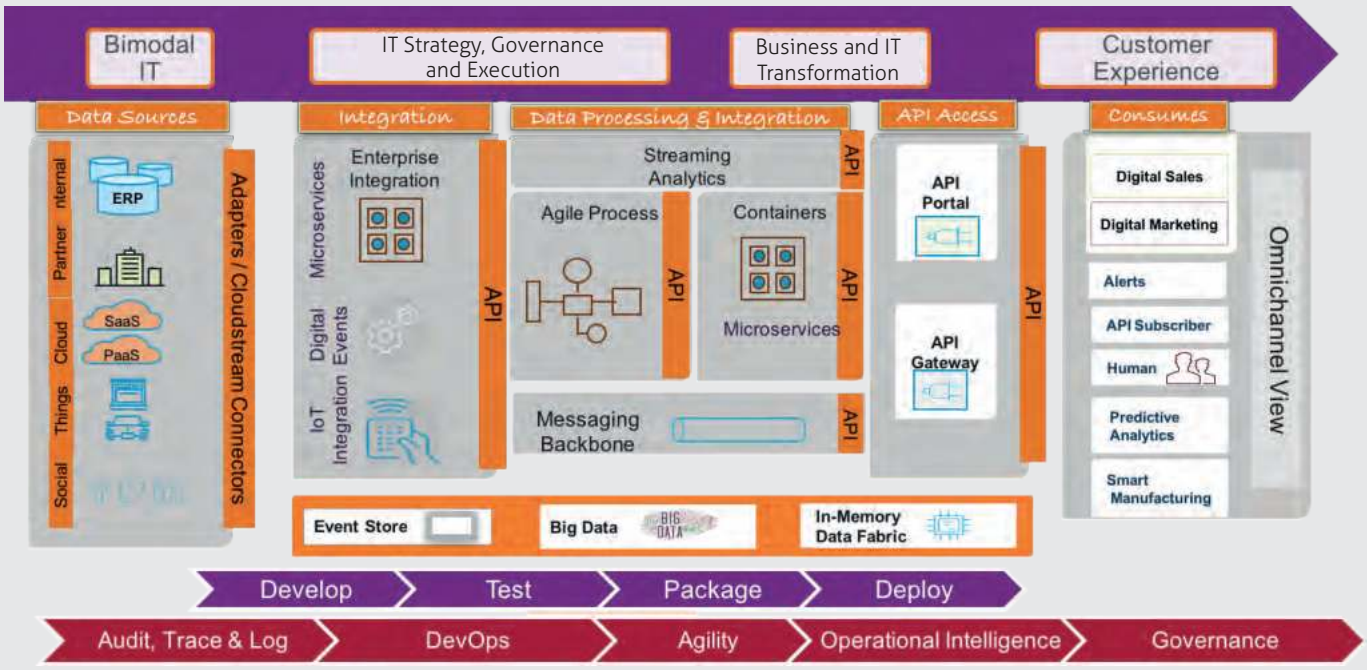
'API First' adoption strategy is practiced by LTIMindtree across partners and customers. Using API as platform it provides tangible benefits in a short to long term API engagements and consultancy services.

API landscape is focused towards three major areas – firstly current offering to API adoption customer (In alignment with API Maturity adoption), second – Strategy execution and thirdly – Market growth and presence realization with API adoption.

## Table shows different kinds of API platform and benefits.

| API Platform | Benefits | API Types |
|---|---|---|
| **Run Business Anywhere**<br>Connect and share enterprise data with Cloud service providers.<br><br>**Single published interface for the services**<br>Used by development, testing, and consumers for integration.<br><br>**Managing Access**<br>User & application management, SSL, registration/access tokens, encryption.<br><br>**Partner Subscription**<br>Manage Metering and Billing of APIs.<br><br>**Hosting**<br>Services hosted in the client's own data center or hosted in Cloud.<br><br>**Caching**<br>Reduce frequent requests to the database server<br><br>**Don't re-invent the wheel**<br>Leverage COTS products for existing capabilities | ■ The APIs support **multiple channels** of access, preferably without the service needing to change its implementation to support other channels<br><br>■ **Increase business agility** and reduce development costs since services can be reused across channels<br><br>■ Hybrid platforms provide strong connectivity with **existing assets** (ERP, CRM, Messaging backbone)<br><br>■ **Secures the communication** with partners against multiple threats and attacks<br><br>■ Measures and throttles the consumption of services as per subscription tier, **SLA, policies**<br><br>■ Promote **reuse** and development of mobile apps through developer portal and device registration. | ■ **System APIs** connect with existing assets (System of Records) through multi protocol, enterprise adapter based integrations. They are built for longer sustainability and relatively complex solutions.<br><br>■ **Process APIs** help in building a hybrid integration model, aggregate information, and route to System APIs.<br><br>■ **Experience or Channel APIs** are lightweight, enable faster integration, suitable for partner subscription management, published for app develop ment, and experimentation.<br><br>■ **Device APIs** are primarily exposed for consuming information from sensor enabled devices through lightweight messaging protocols.<br><br>■ **Public APIs** are consumed by end users and clients<br><br>■ **Private APIs** are consumed by internal systems and channels<br><br>■ **Partner APIs** are consumed by B2B partners in the ecosystem<br><br>■ **API Marketplace** are new revenue generating avenues using APIs |

## An example of our CPG Reference architecture where "API First" approach is adopted starting from conceptualization to realization phases of project execution.



ECS integration practice and delivery teams supports and aligns combined strategies for API management, PaaS, and Hybrid Integration. Entire bundle extends solution spans across unified solution for API management, APIs built using Node.js standards, and developer focused API development and deployment.

## LTIMindtree's API Adoption and Practice Focus Areas

- API implementation functions based on semantics, modules, and functions

- Tighter connections with systems of record

1. Offloading legacy, mainframes, plm, crm, and erp data in the form of data services considered as BaaS 'backend as a service'
2. Custom built-in libraries and functions exposed as BaaS APIs as adapters
3. CRUD operations performed as write-behind pattern

- Adding features to digital practices and projects like –

1. cognitive computing,
2. big data,
3. machine learning
4. legacy integration
5. streaming analytics

- Supporting industry standards data models like financial services, healthcare, air traffic management

- High level business solution focus

- B2B use cases

1. Federated Identity management
2. Features for administrators at B2B partners
   i. Manage respective API users and teams
   ii. Serve internal API scenarios
3. Full featured API gateway
   i. Policy authoring
   ii. Quotas and rate limits
4. Open web APIs
   i. Package custom API policies – used by API management tools
5. API product definition
6. Digital business ecosystem vision
   i. Unified B2B relationship
   ii. EDIs and MFT
   iii. Serve huge transaction volumes from IoT, Mobile, and other digital sources

- API mediation and performance

1. Paging, caching and message enrichment
2. VETRO Pattern
   i. Transform, route, and mediate (SOAP ↔ REST, XML ↔ JSON)
   ii. Message parsing, validation, translation, and enrichment
   iii. Service aggregation, virtualization, and orchestration with simulation

- Data center consolidation and migration with respect to middleware (Common integration platform)
   i. Deployment models extended to hybrid in addition to SaaS based and customer managed

- API lifecycle management

1. From inception to implementation to retirement
2. Versioning (how different versions of the same API can be leveraged to meet rapidly changing business and customer needs)

- API solution security

1. Security integration
2. Closed-loop OAuth2 – Mitigate complex enterprise scenarios through configuration rather than custom coding
3. Device-to-back-end API authentication
4. Device-level certificate management
5. Role based access control
   i. Define usage quota and limits by application (roles and groups)
   ii. Traffic throttling and shaping
   iii. Content routing and blocking
6. Single sign-on to multiple applications
7. Transfer user sessions across devices
8. Flexible portal customization
9. DoS – Denial of Service protection and pre-notification
10. API attack protection
11. Bot detection – preprocessing, comparison of feature code, API HOOK, fast detection, and sequence detection
12. Message and Transport layer security (not limited to digital signatures, message envelopes, and encryption)

- API economy

1. Integrate pricing and billing out of the box ⊠ direct revenue model from APIs
2. Quotas and rate limits
3. Customized API portal

- API analytics

1. Streaming APIs
2. Monitor SLA and QoS
3. Service profiling
4. Problem identification including guidance in debugging
5. API usage tracking and trend analysis (API retirement consideration)
6. Audit trails

- Configurable lifecycle management

1. Lifecycle stages and transitions
2. Approval workflow based requirements

- Federated LOB publishing

- Policy authoring, processing

- IoT scenarios like – connected cars, telematics, smart logistics, medical devices, industrial IoT

- Integration with DevOps tools

1. Live API creator tool
2. Microservices
3. Containers
4. Continuous integration
5. Configuration management
6. Integration with PaaS
7. Integration with processes and events platforms

- Development, Collaboration and Governance

1. Team publishing
2. Driver of Agility
3. Composable enterprises
4. Web based development tools
5. Service life cycle governance
   i. Effective operation of mature
   ii. Communities features
      1. Co-creation processes with internal and external API users

- API socialization and deep collaboration

i. Self-registration and subscription
ii. Access to documents, schemas, models, services based on level of authorization
iii. Blog, ratings, and comments
iv. Incident ticket management
v. Social media integration (followers and RSS feeds)
vi. Promote and test services

- Open source API platform

1. Identity management
2. Integration (Mediation, ESB, Streaming analytics, Operational Intelligence, IMDF, Policies, Audit, DevOps, and Governance)
3. Open standards adoption for federal identification and authentication using OAuth and SAML
4. Process management and orchestration
5. Event driven architecture
6. API manager product package
   i. API users
   ii. API publishing and implementation
   iii. Team publishing
7. Service lifecycle management
8. Reports (Based on filter conditions)
9. Back end API orchestration availability in SaaS
10. Public enhancement API backlog
11. Hybrid architecture adoption
    i. Multiple traffic delivery mechanisms including open source
    ii. Integration with caching and distribution power of full CDNs
    iii. Unlimited fully hardened gateway nodes
    iv. Deployed multiple kinds of traffic delivery from vanish gateways to global CDNs, vendor specific APICast Cloud proxy, all the way to lightweight code plugins
    v. Anonymous functioning to deliver traffic while synching with the core management layer out-of-band
    vi. 'Mix and Match' based on traffic pattern

- Industry specifications, adherence, API marketplace

1. RAML spec during design and definition of APIs
2. Swagger specifications – Import and export Open APIs
3. Integration with Apache camel – Combined API management and implementation platform
4. Trello – API project management and organizing assets, artifacts
5. Access varied arrays of APIs through API marketplace – adoption in progress
   i. Consolidation of APIs from varied vendors into single account
   ii. Slimmed down API management platform

Hybrid integration platform brings in APIs lifecycle management, unified, and centralized runtime service monitoring approach during design and implementation phases.

## LTIMindtree's GTM repository

API repository consists of accelerators, tools, artifacts, and POVs

- Templates
  - Digital Integration Architecture Assessment
  - API Architecture Assessment
  - Digital Integration Product Evaluation Framework
  - Hybrid Integration Platform Adoption Assessment
  - DevOps Maturity Assessment
  - API Maturity Assessment
  - API Design Documentation Template
  - Digital Integration Documentation Template
  - BPM Technical Design Documentation
  - API Deployment Model Design Document

- Methodologies
  - API Lifecycle Management
  - API Adoption
  - API Monitoring – Data Lake view
  - API Operational Intelligence
  - Unified and Centralized Management/ Administration and Monitoring

- Tools
  - Distributed caching,
  - Traffic throttling,
  - Quota limit, billing rate, billing fulfillment,
  - Pricing,
  - Automated API retirement notification,
  - Logging framework,
  - API services profiling,
  - Test automation,
  - Exceptional handling, audit trail, and trace mechanism

- Use Case solutions and frameworks
  - OAuth setup
  - Product and Price comparison
  - API framework for Apigee and SAP ERP
  - Scaling and aligned with Hybrid Integration

## API Takeaways and Challenges

- Rapid Growth of the API industry accelerates even more with mobile and IoT expected to drive future growth
- Improved integration drives decision making as lack of integration between existing tools holds teams back
- Organizations understand the need for API security and want to see the security challenge solved in the years ahead
- API performance is paramount - for both providers and consumers
- API provides struggle to balance speed of delivery and API quality
- API provides face losing users/customers as a result of quality issues
- The demand for efficient, easy-to use tools increase as Teams try to manage limitations on time, resources, and skills
- The lines between teams blur as developers, testers and IT/Operations are involved throughout the entire API Lifecycle
- Teams understand the need for standardization but face challenges when implementing key components

## LTIMindtree's ECS Practce Straregy Summary

**LTIMindtree's API GTM enabling offerings is driven by custom and hy**brid approach taken along with product vendor and CoE group teams in ECS practice.

- Integration between application integration, data integration, process integration, and API management is achieved by identifying gaps in product components by providing home grown frameworks and solutions.
  - Enables different roles to work together across citizen integrators, Ad-hoc integrators, and integration specialists
- Practice supports the integration of mobile apps, B2B, IoT, data from social media, and cloud deployment models
- Practice multiple deployment models like – cloud, on-premises, and hybrid
- Framework around unified and centralized management and administration and monitoring for different technology vendors
- Adherence to Rapid application development requirements based on exploiting leveraging rich set of pre-packaged integrations, cloudstreams, and adapters to improve overall productivity and supporting high velocity digital business requirements
- Managed active ecosystems of software vendors, technology partners, and customers
- ECS Integration practice team formation is a result of continuous journey for more than a decade
  - Socialized HIP capabilities with LOB stakeholders
  - Timely identification of outcomes and contingencies, mitigating risks
  - Evolving nature of capabilities yielding productive solutions
- DIY or Self-service integration practice for customers involving different integration stakeholders
  - Integration facilitation team as part of CoE
  - iPaaS and iSaaS based deployment framework practice
- Measure tangible value of integration deliverables (API first approach)
  - Productize foundational APIs with APIs economy
  - Reuse these assets across organization
  - Integration products value equity attached to measurable objectives
- Cost reduction
- Process improvements
- New revenue opportunities
- Improved customer experience

Debashish Maity is a recognized expert in the enterprise integr ation. He brings over 13 plus years of experience in the presales, development, architecture and design, and implementation of large scale integration and business transformation programs. He hasIT experience spanning over API, SOA, BPM, DevOps, PLM, MDM, EA Practice, ERP, and Data Science. He leverages his global experience to bring unique perspective on the business transformations across industries including manufacturing, CPG, Telecom, Banking and Financial Institutes, and Consultancy.

Debashish holds a Master from Symbiosis Institute of Computer Studies and Research, Pune India.

*Email – Debashish.maity@ltimindtree.com*

**LTIMindtree** is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 700 clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by 84,000+ talented and entrepreneurial professionals across more than 30 countries, LTIMindtree — a Larsen & Toubro Group company — combines the industry-acclaimed strengths of erstwhile Larsen and Toubro Infotech and Mindtree in solving the most complex business challenges and delivering transformation at scale. For more information, please visit **https://www.ltimindtree.com/**