



Modern Architectural Paradigm for Digitization

Practice Head:

Srini Peyyalamitta

srini.peyyalamitta@aspiresys.com

Author:

Deepthi Jayaram

Solutions Architect - Product Engineering

deepthi.jayaram@aspiresys.com

aspire 
SYSTEMS
attention. always.

C O N T E N T S

- Appetite For Modernization
- Transforming To A Modern Architecture
- Digital Transformation
- Modern Architecture Sample
- Conclusion

Modern Architectural Paradigm for Digitization

Technology has found its way into every aspect of our life making the world around us “smarter”. Software plays a driver role in creating the smartness for technology.

We frequently see new software being released almost in parallel to new versions of existing software. Every once in a while, some software takes the technical world by storm but many a time that success is not long lived as we have seen very popular software go into near oblivion by the arrival of a new and better counterpart meeting the necessities as it emerges.

So, a software success is not truly decided by its technical sophistication of today but its ability to adapt to new technological needs of future. Survival of the fittest has taken on a new meaning in the technical world where, software (architecture) modernization is not a luxury anymore but essential to meet the necessities of software agility as it evolves. Hence, businesses should always have a finger on the current technical pulse in order to make a decision on when, why and how their product should be modernized.

Appetite for Modernization

Some of the common questions asked on Software Architecture Modernization are “When should I start thinking about it? Why is it required now? How will it improve my business? When will I start seeing the ROI? ”

Appetite for software architecture modernization generally varies with businesses depending on their customers, geographies, growth phase and competitions at large. Gartner predicts that long-term growth in spending will be driven primarily by modernization, functional expansion and digital transformation projects. According to this citation of IDC, two-thirds of CEOs plan to focus on digital transformation strategies for 2016 and that CIOs will be major players in leading every department through this shift.

The typical business scenarios that warrant modernization are

1. Current software is lagging in aspects like performance, scalability, security and flexibility for growing user demands
2. Deeper expansion into current domain – Horizontal or geographic expansion
3. Venture into new avenues – Moving the business into different sub-areas in its existing domain
4. To be prepared to tackle unexpected market demands
5. Connecting to the users via social media channels.
6. Current legacy software need to be exposed as services to integrations
7. Predictive and relevant delivery to customers

Every organization, irrespective of its size, will fit into one or more of the above categories. In order to survive and be prepared to face the future, it is better to make the changes in small installments gradually than to have to rush into big bang modernization suddenly due to some pressing business demand.

We recently had a scenario where a market leader in providing K12 educational content was not able to predict their LMS sales drop. This drop was because the schools no longer wanted to use multiple applications. The organization then decided to invest in upgrading the architecture of their LMS products in an incremental way. As a result, their services and data were exposed directly to the LMSs using a token based approach. This modernization approach eventually helped in increasing their sales figures.

Modern Architectural Paradigm for Digitization

All software products that are future ready are expected to possess these characteristics:

- Lesser time to market advantage (to be able to build incrementally and quickly)
- Ease of integration with third party tools and libraries
- High scalability
- High performance
- High availability
- Easy to maintain and expand
- Flexibility

These technical requirements are being made easier by the day with new frameworks and modern day solutions. The most intelligent step then, is to have a product architecture that is ready to make use of these innovations and lead to tangible business benefits.

Products with modern architecture can be achieved through a simple solution of directly adopting a framework or a very complex solution of building a custom framework. Most cases can be covered in the first half of this range if we use the right tools the right way. Every technology in our stack should have a clearly defined purpose and should be used in the best possible design so that the business reaps both the short-term and long-term benefits.

Transforming to a Modern Architecture

Some key points to be considered for transformation are:

1. We need to be able to incorporate new business features and enhancements while building the modern architecture.
2. We should be able to support the monolith code base until the new system is completely ready.
3. We need to port customers to the new system.
4. We need to support specific customization that was done for users in the monolith system.

5. To transform a monolith system which is too tightly coupled to the front end layer, we need to break it down, reevaluate the business needs and provide provision for backward compatibility.
6. To transform a system that has acquired a high technical debt, it is better to take this transformation as an opportunity and re-design it from scratch.
7. We need to make sure that the current deployment set up is scalable enough to support the new architecture.

The best way to transform to a modern architecture from a monolith is an incremental approach. This is more cost effective and will also enable a smoother transition. The customer and the business will also realize the ROI faster.

The aim would be to build a basic framework with the latest technologies suited for the business and to incrementally move small pieces of the current product into the new framework. The aim should be realistic, achievable and should have measurable goals and sub goals with a defined time-line as it is easy to deviate and get carried away in trying to build something which is technically great but not in line with the business.

If there is an opportunity, the product should be taken to the cloud. One key point here is to keep the changes small and to use as many off-the-shelf solutions available so as to not have a negative impact on the current business. In other terms, we should absolutely not “re-invent the wheel”.

Digital Transformation

One of the most important reasons for modernization is digital transformation. Everything and everybody is digital now. People want to be able to access data across multiple applications from the comfort of their homes on ever evolving laptops, tablets and mobile phones. They want to be able to connect their applications to the latest social media and be able to use common accounts like google, Facebook etc. to connect to applications.

Modern Architectural Paradigm for Digitization

Organizations are relying on Chief Digital Officers for strategy building, innovation and relationship building. The trend is also shifting towards blending software development and operations roles and adopting the so-called DevOps approach to simplify the creation and deployment of new applications faster.

Hence, the following aspects have become integral for any product to survive in today's world.

➔ Social integrations

Social integrations like Google, Facebook, Twitter, LinkedIn, RSS Feeds etc.

➔ Security

Security against common and domain specific cyber threats

➔ Standards and compliances

Standards and compliances should be in place especially to ensure the confidentiality, integrity and availability of personal data

➔ Mobile compatibility

Now almost every site is made accessible on different types of mobiles and tablets. Designing UI which caters to all of these devices is a mammoth challenge in itself. To be able to implement and maintain it is a bigger challenge. But there are some frameworks like Bootstrap, foundation, UXP etc. which helps in overcoming this challenge. These frameworks have all the basic settings and have the flexibility for much more.

➔ Analytics

The analysis of user behavior in order to predict business directions and decisions is becoming more of a reality every day. To collect and store usage related data is the first step towards this but transforming the data to a usable form is the key. Many tools like Heap Analytics, Inspectlet, Mouseflow etc. provide services to capture every user interaction which can be used for analysis and reporting. This analysis can be used to improve the user experience drastically.

It can even be used for business prediction and prescription using tools like R, Apache Mahout, IBM SPSS, Mathematica, Oracle Advanced Analytics etc.

➔ Cloud

High accessibility, flexibility and scalability are some of the most popular reasons to go to the cloud. Maintenance of hardware and related issues will be a thing of the past. Pre-defined back up will be available in terms of business contingency plans. Ease in development and deployment act as cherry on the cake.

With cloud development, teams can expand and contract as per business needs. Transition time between team members is highly reduced. Moving to latest technologies will practically have no impact as far as development systems go as they will be centrally changed and reflected across all development systems thus making the transition very manageable. This is like a dream come true for organizations which want to focus more time and energy on their business.

With cloud deployment, teams will be able to push fixes and features to production at a faster rate. Automated testing can help automate the whole process leading to a deployment model where developers need not get involved at all. This will increase productivity and help keep the developers' focus on quality delivery of code.

➔ Customer Experience Management

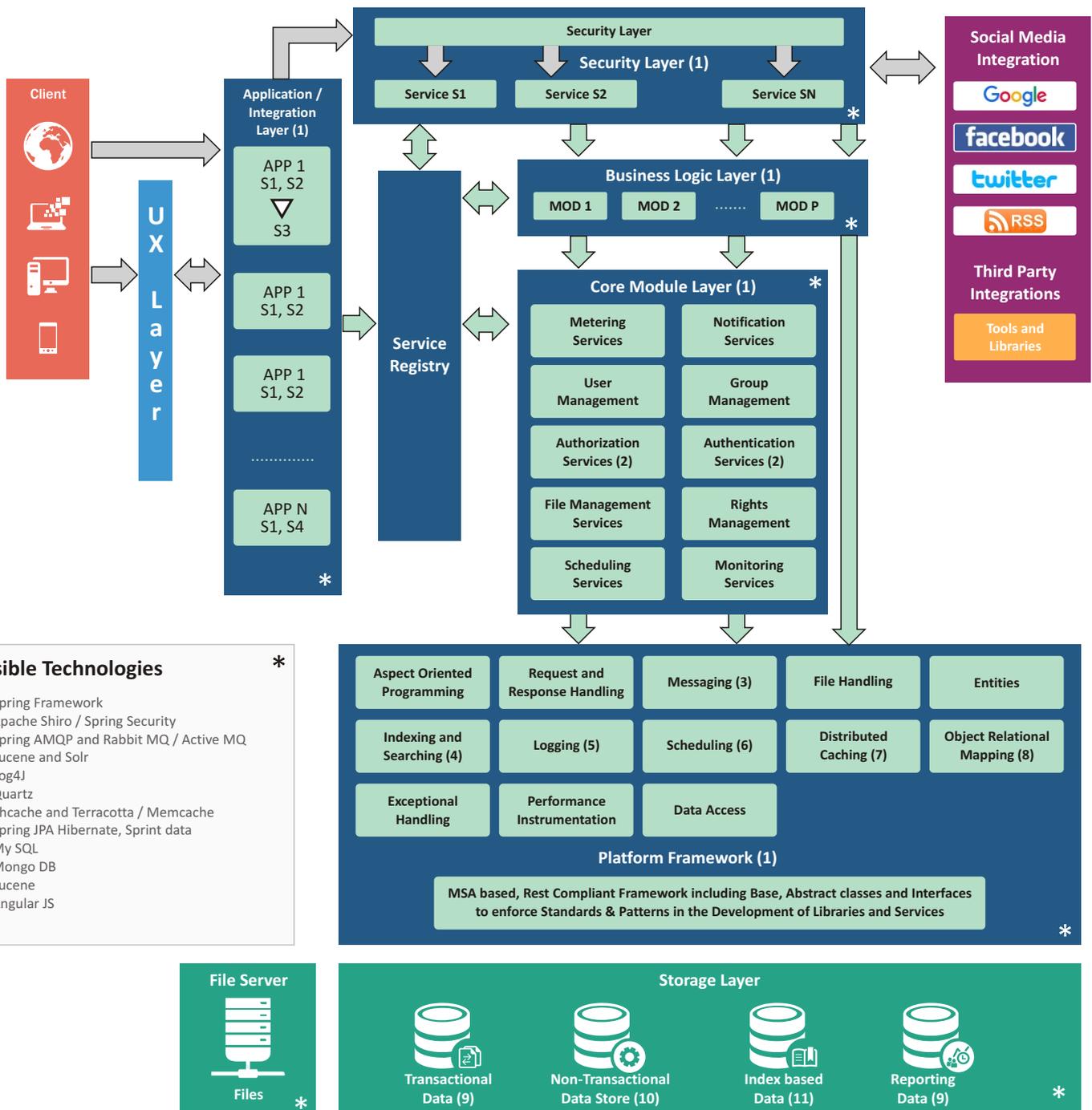
CEM includes

- ➔ Delivering consistent, positive customer experiences
- ➔ Seamless delivery of the product into multiple form factors.
- ➔ Personalized themes and preferences
- ➔ User aware and device aware layout
- ➔ Information architecture – Domain based representation
- ➔ Ease of use of the product
- ➔ Eye catching and engaging visual effects
- ➔ Instant collaboration

Modern Architectural Paradigm for Digitization

Modern Architecture Sample

The following is a Microservices based Architecture design which can make the application ready for any business need. The modules are split up to serve different business purposes and can be deployed individually along with the core and related software if required. This module based approach is easy to build, flexible to deploy and easy to maintain.



Modern Architectural Paradigm for Digitization

➔ Storage Layer

Storage Layer comprises of all transient and persistent data schemas and structures used to store the information. The layer is responsible for data management without provision for any business logic. The layer is responsible for supporting concurrent reads and writes with complete support for ACID. The layer supports scalability by providing for Horizontal Partitioning and support for Clustering.

The store comprises of the

- ➔ Relational Data store (Transactional Data) - for storage and handling of structured data. The store is used for durable storage of data in an organized fashion for transactional purposes. The structure is defined by the schema for the databases.
- ➔ Non-relational Data store (Non Transactional Data) - A document based store is used for handling highly accessed, non-structural data. The use of big data stores is becoming very popular where fast access, storing huge amount of data and flexible schema is required.
- ➔ Search-based Data store (Index Based Data) – A search-based store is used to maintain data in a convenient format such that it can be searched easily. The store should perform real-time indexing so as to maintain the integrity of data searched. It has a little less flexibility than big data stores in terms of schema but can support customized algorithms for searching text based data.
- ➔ DataMart (Reporting data) - A de-normalized data structure is used as the scaled down data warehouse to store all historical information based on the above data structures. This data structure is used for all historical and trend analysis of the data in the system. This can be also be used for Business Intelligence Reporting. This can be stored in either relational or non-relational data stores.

➔ File Server

File Server is a dedicated, third party, secure server used as a shared file store. It enables storage and retrieval of files such as text documents, images, audio, video etc.

➔ Main Framework

Main Framework is a MSA-based, REST-compliant framework that aims to enforce standards and patterns while developing new libraries and services. These libraries are also used to provide third party integration.

Together, they provide the base for:

- ➔ Connection to the relational, document-based and index-based data stores as required.
- ➔ Parsing of XML or JSON or Form Elements as input data and building of XML or JSON output data through Data Transfer Objects (DTO).
- ➔ Returning HTTP response codes.
- ➔ Returning generic information for every request.
- ➔ Handling various generic exceptions like Data Store connectivity exceptions, data access exceptions, input and output parsing exceptions and return the appropriate error code and http response code in the response.
- ➔ Handling distributed caching.
- ➔ Sending to and receiving messaging from a queue.
- ➔ Logging information for debugging.
- ➔ Creating and scheduling jobs.
- ➔ Capturing of performance data, pre and post processing a request.
- ➔ Capturing analytics based on usage and experience.

➔ Core Module Layer

Core Module Layer is a basic module comprising of non-domain specific services. These services may optionally be used across modules having different business concerns. This module reduces redundancy and increases consistency across modules.

Modern Architectural Paradigm for Digitization

Some of the services provided by this module are as follows:

- Create, Retrieve, Update or Delete
 - User
 - Group (of Users)
 - Rights (Access Permissions for users/groups)
 - Files
 - Emails
 - System Messages
 - Jobs
- Performance Measurement
- Auditing
- Authentication (Identifying who is accessing)
- Authorization (Identifying what can be accessed by the authenticated user)

→ Business Logic Layer

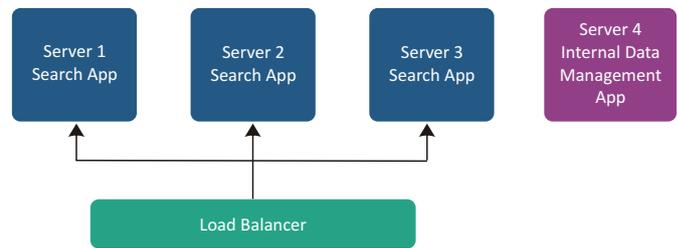
This layer comprises of reusable modules where a module is a set of related business logic. A new module is built on top of the Framework and references one or more platform libraries. It may also reference the core module for one or more of its business logic.

→ Service Layer

This layer comprises of services performing some specific tasks. A service comprises of two or modules which work together to provide a business goal. Building an application using this architecture should be fast and reliable thereby decreasing the time-to-market for new and enhanced applications.

→ Integration / Application Layer

This layer exposes different applications to the public using a standard REST API. It may contain one or more services and may optionally use services in different combinations (service orchestration). The advantage here is that some specific applications/services can be deployed across servers in order to support more load as compared to say a backend app or an internal data management app.



→ Client

The client can be templates like FTL, JSP, PHP etc. or can be application built using Angular JS or MOBILE, JAVA, .NET, C++ etc.

→ Digitization and Integration

The layer allows the application to easily and seamlessly integrate with social platforms and third party systems. It aims to have a one stop experience for a user wanting to navigate between different applications in a device of his/her choice. These integrations should be flexible enough to evolve and adapt to the latest tools and devices.

Focus for any organization today should be on mobile devices and apps, cloud services, mobile broadband networks, Big Data, and social technologies.

Conclusion

So far we have discussed on the business and technical reasons to go for architecture modernization. We have also touched upon the key points to be considered for transformation and a suggested approach. We have then seen a sample modern architecture in great detail. This architecture encompasses all the features required for a state-of-the-art micro-services based product with a strong foundation, latest social media integration, security layer, responsive UX layer, flexibility, scalability and the ability to be deployed to cloud. It can be applied in any given domain and will play well with an incremental transformation approach.

Architecture modernization is without question the path to the future. It will matter greatly as to when an organization undertakes it.

Modern Architectural Paradigm for Digitization

The most important factors which will decide the success and failure of this transformation are the team involved, the approach taken and new architecture designed. The best team is one which keeps the business priority at the top at all times but does not compromise on the technical aspects. The best transformation approach is an incremental one with defined sub goals and strict timelines. The best modern architecture is one which caters to all the current and future business needs with just enough flexibility to take on the unknown.

References

1. <http://www.gartner.com/newsroom/id/3119717>
2. <http://www.cio.com/article/3009670/it-strategy/digital-transformation-will-shape-2016.html>



Aspire Systems is a global technology service firm serving as a trusted technology partner for its customers. The company works with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in Aspire's specific areas of expertise. Aspire System's services include Product Engineering, Enterprise Solutions, Independent Testing Services, Oracle Application Services and IT infrastructure & Application Support Services. The company currently has over 1,600 employees and over 100 customers globally. The company has a growing presence in the US, UK, Middle East, Europe and Singapore. For the seventh time in a row, Aspire has been selected as one of India's "Best Companies to Work For" by the the Great Place to Work® Institute, in partnership with The Economic Times.

SINGAPORE
+65 3163 3050

NORTH AMERICA
+1 630 368 0970

EUROPE
+44 203 170 6115

INDIA
+91 44 6740 4000

MIDDLE EAST
+971 50 658 8831

For more info contact
info@aspiresys.com or visit www.aspiresys.com

