

SOFTWARE ARCHITECTURE ASSIGNMENT 2

E-Commerce Product Based Application (EPBA)

Amit Naik
2022mt93215@wilp.bits-pilani.ac.in

About

EPBA application is e-commerce based application with fully dynamic based in adding product and able to change theme including company name and logo.

Application is built on microservice and micro-frontend based architecture with all dynamic functionality

Purpose of the EPBA System (Goal)

Below are the purpose of EPBA system

- ❖ People to buy and sell physical goods, services, and digital products over the internet
- ❖ Business can process orders, accept payments, manage shipping and logistics, and provide customer service.
- ❖ Application is fully configured and can be configured to any company name and logo
- ❖ Application can change theme and color based on any company's and used as product

Key requirements of the system – functional

- ❖ Fast, stable & secure web hosting.
- ❖ Highly-responsive mobile site.
- ❖ User-friendly web design.
- ❖ Robust operation systems.
- ❖ Fast checkout process.
- ❖ Transparent brand information pages.
- ❖ SEO-friendly web pages

Key requirements of the system – non-functional

- ❖ Performance – Homepage should load in less than 4 seconds on iOS 10+, Safari on 4G and any browser
- ❖ Security –
 - ❖ System should have clear all cyber security related issue and clear OWASP vulnerability
 - ❖ System should have role base authentication and authorization for user logged in
- ❖ Availability –
 - ❖ Recovery time should be less then 30 seconds
- ❖ Usability –
 - ❖ Should be workflow configuration, where it should reuse in other sites like Airport or military
- ❖ Interoperability –
 - ❖ System shall abstract common services such as logging, remote communication to ease flow development for flows

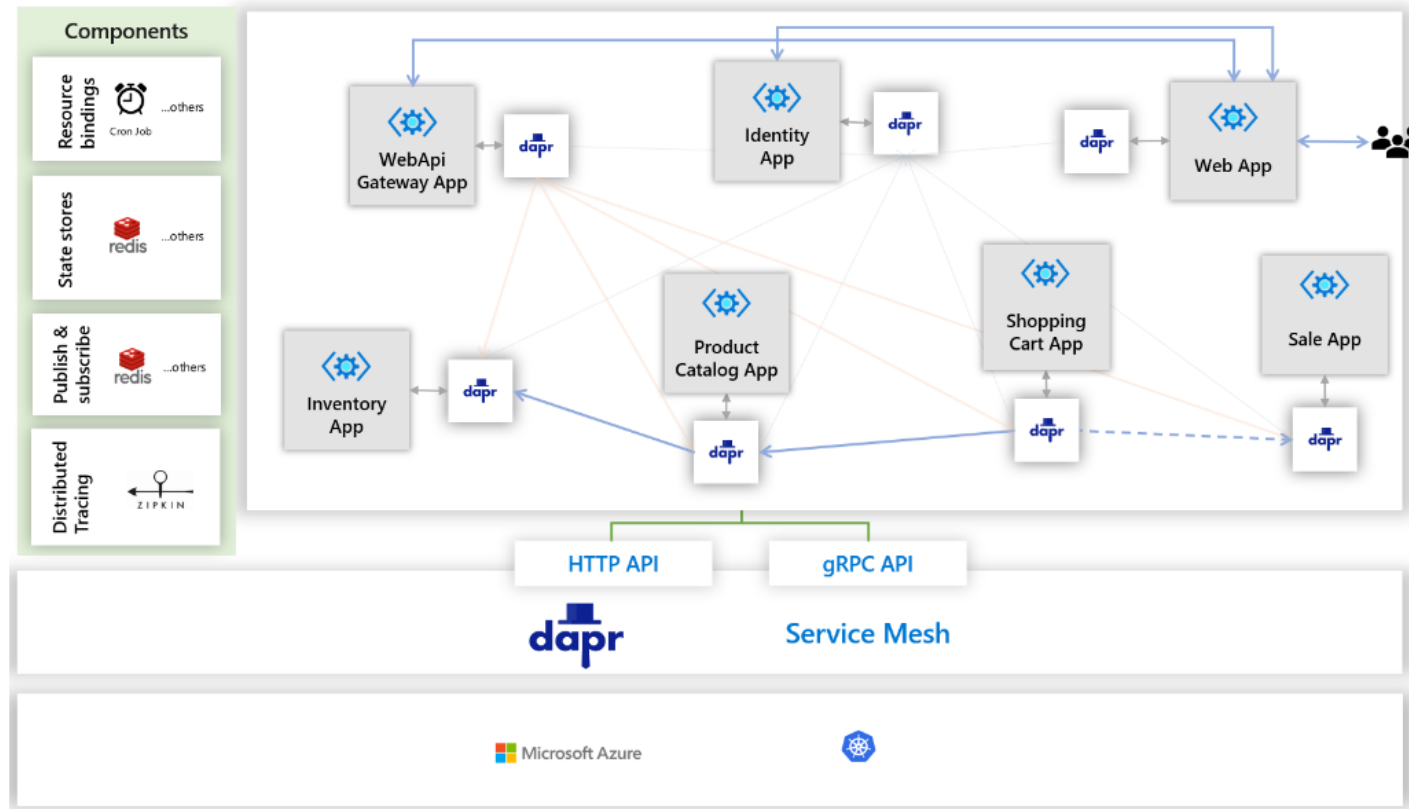
Utility tree of Architecturally Significant Requirements (ASR)

Quality Attribute	Attribute Refinement	Scenarios	Business Value	Architecture Component
Performance	Scalability	System should be able to Scale in/out based on server load	M	M
	Response time	Should be less then 10 sec	H	H
Security	Authorisation	System should have role base authentication and authorization for user logged in	H	H
	Confidential	Vehicle number and personal detail of customer should be encryted and stored in database	H	M
	Operation and maintence	Should log all the exception	M	H
Interoperability	Unified experience	System shall abstract common services such as logging, remote communication to ease flow development for flows	M	H
Maintainability	Maintence	Monitoring or Health check interface should be provided to check all service is up and running	M	H
	Adaptive	Should be proper decouple with database so that migrate to other database	M	H
Usability	Learnability	Provide proper documents and FAQ documents	H	H

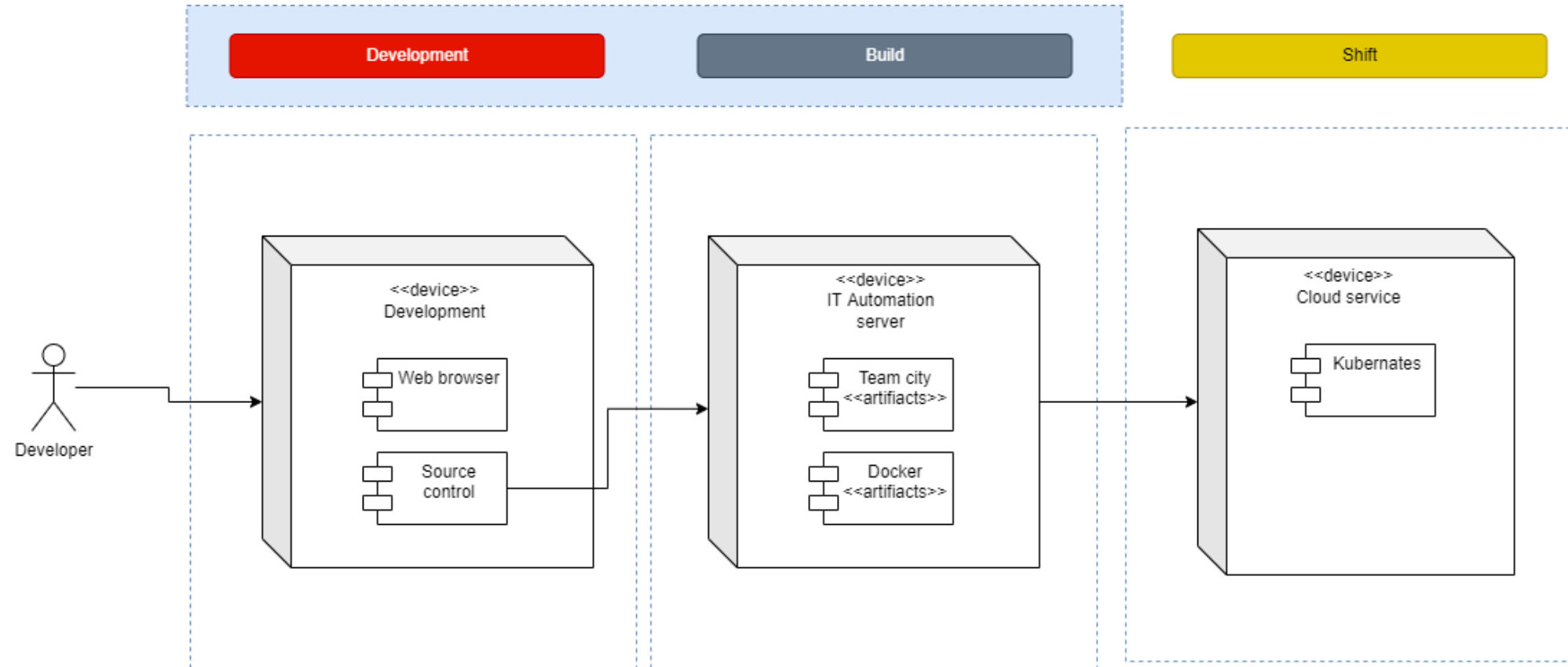
Tactics to achieve the top 5 ASRs

Quality Attribute	Scenario (Attribute refinement)	Tactics
Performance	The system interfaces and works with existing popular smart agriculture systems that use standard protocols for communication. .	Control hub that maintains the smart agriculture network must support popular smart home communication protocols such as ZigBee, Bluetooth LE or Z wave
Security	Only authorized users can interact with the smart Agriculture system.	Users will get access to the system only after they have done their authentication with SSO. Secondary methods such as SMS/OTP authentication can also be used for authentication purposes.
	The systems within the smart Agriculture network should use encryption supported protocols to make sure the communication channels are secure	Any communication between smart systems must be encrypted with the network key or other tokens created while authenticating the user
Maintainability	Detect faults with any systems in the smart Agriculture system environment.	The controller hub checks for the availability of the systems by employing responses/messages. When there is a failure in response after a fixed number of retries then fault recovery can be triggered (This can be done activating the backup redundant system). The hub can also notify the user of failures in the system. .The application should scale up on demand and not fail due to faults related to high traffic
	Recover faults when it is detected in any system or Controller hub within the smart Agriculture system	Backup systems should be readily available if the system fails The application hosted would also need to employ redundancy to make sure cloud services are highly available and use deployment strategies which aid in scaling up on demand

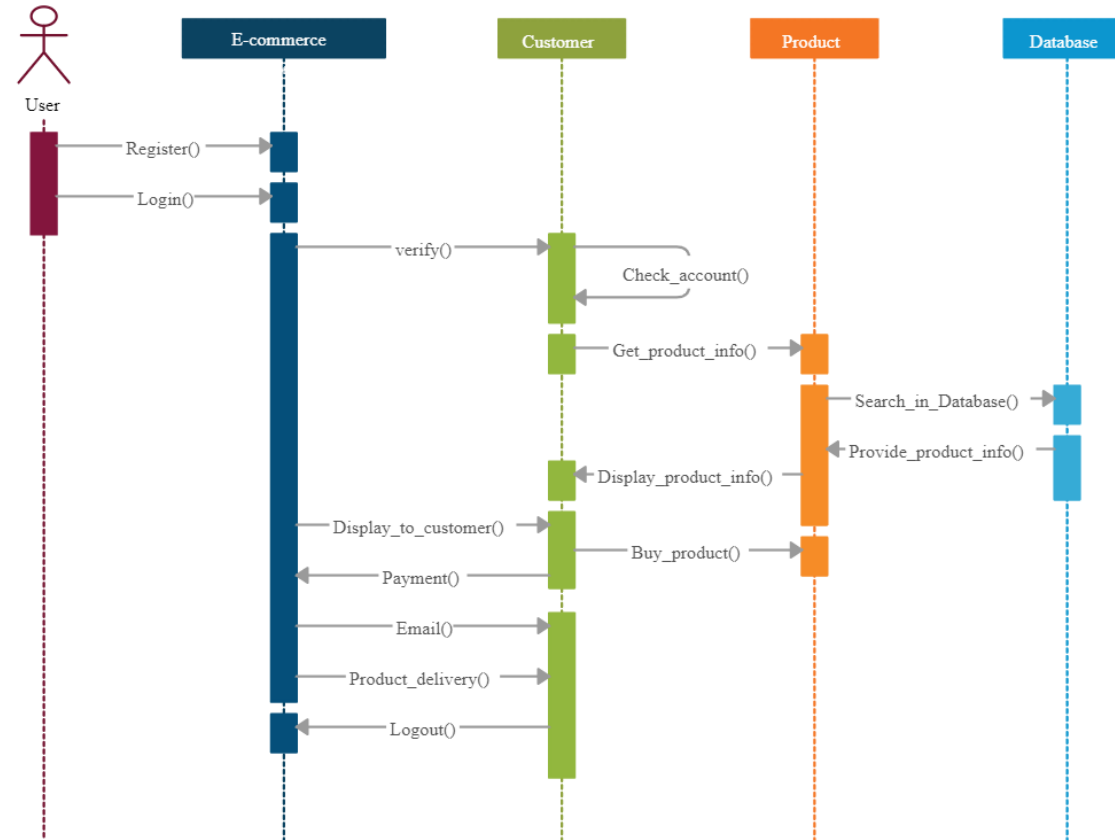
Software Architecture diagram – Component & Connection diagram



Software Architecture diagram – Deployment diagram



Software Architecture diagram – Sequence diagram



Architecture Patterns Used

❖ **Microservice Architecture**

- ❖ Microservices improve performance because teams handle specific services rather than an app as a whole. This advantage leads to greater accountability, compliance, and data security. Microservices allow developers to become language and technology agnostic.

❖ **Micro frontend Architecture**

- ❖ This is used from UI related
- ❖ Micro frontends are especially useful when we have projects with dozens of developers working together in a large business domain and want to reduce complexity by splitting into multiple subdomains, independently deploying different parts of applications without overloading communication and coordination between teams

❖ **Publish Subscribe Pattern**

- ❖ To make sure that events interact independently with each system we make use of Publish subscriber pattern
- ❖ In the publish-subscribe pattern, Systems interact via announced messages/responses whenever an event occurs. Systems may subscribe to a set of events for example if the temperature is set at a particular degree to maintain and if the temp exceeds then, the system sends response for an event to occur hence notifying the user through the app

Key learnings

- ❖ Will get clear understand on non functional requirement that can potentially change the way system is modelled.
- ❖ Context diagram helps all stakeholders understand bigger picture of the system and helps showing what belongs to system
- ❖ Component connection diagram helps developer and scrum master visualize implementation strategies much earlier in system designing and makes planning (especially in AGILE) much easier
- ❖ Deployment view gives a clear image to operation teams on how the application modules are deployed and helps in rough estimation of cost early on easing planning