

SOFTWARE ARCHITECTURE ASSIGNMENT 1

Gate Management System (GMS)

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

About

GM interface with third party systems like UVSS, LPR, X-RAY and provides a solution to control the entry of vehicle in site.

A live interactive GUI helps operator at gates to approve/reject entry of vehicle based on disposition received by operator from Gates

Purpose of the GM System (Goal)

Below are the purpose of GM system

- ❖ Live tracking of vehicle in premises (might be in airport premises or military campus)
- ❖ Provided GUI which helps user to approve or reject vehicle in gate based on disposition detail received from hardware devices
- ❖ Should be interacting with hardware devices like LPR, UVSS and X-Ray
- ❖ User can see image w.r.t Vehicle number when required
- ❖ Audit trail to understand, who has given access to the vehicle
- ❖ Able to get report based on required dates and time

Key requirements of the system – functional

- ❖ Live data should be displayed in system and able to track in premises
- ❖ User should be approve or reject vehicle in gate based on disposition detail received from hardware devices like LPR, UVSS and X-Ray
- ❖ System should save all vehicle disposition details with XRay, UVSS and LPR image detail w.r.t vehicle number
- ❖ System should get report based on required date and time
- ❖ System should able to track vehicle entry and exit time

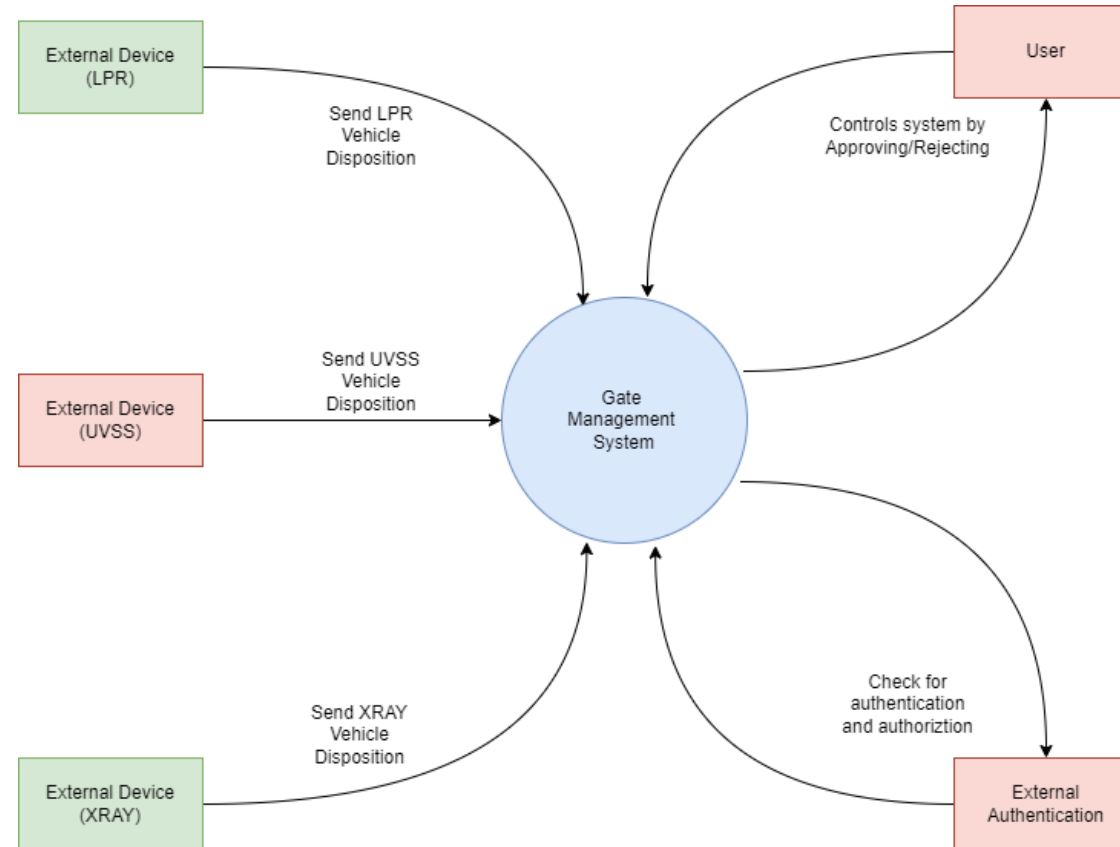
Key requirements of the system – non-functional

- ❖ Performance – As live data should be displayed in system –response time should be milli seconds
- ❖ 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

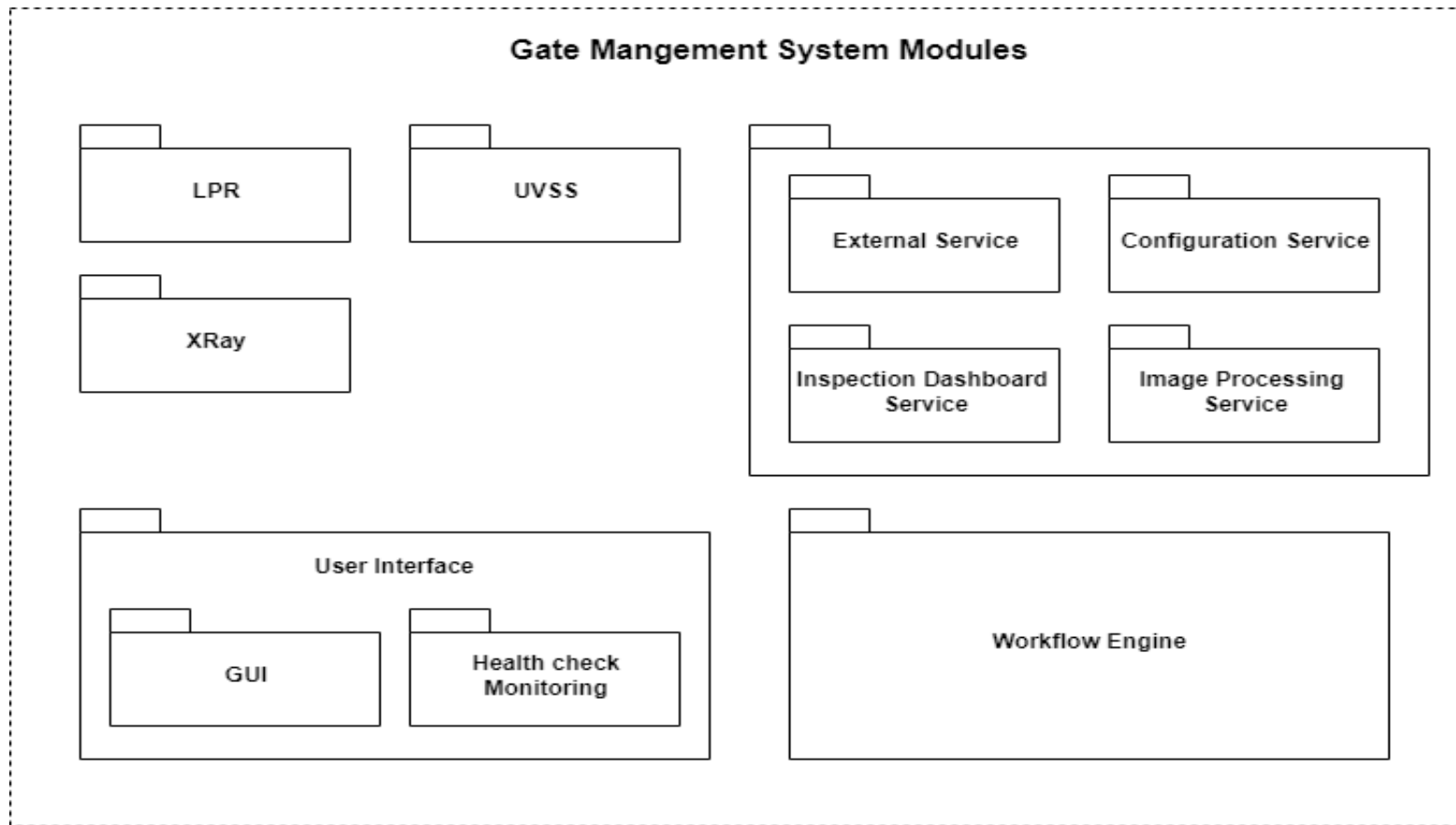
Utility tree of Architecturally Significant Requirements (ASR)

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 3 sec	H	H
Security	Authorisation	System should have role base authentication and authorization for user logged in	H	H
	Rate limit	Should should not allow more then 30 request per sec	M	H
	Confidential	Vehicle number and personal detail of customer should be encryted and stored in database	H	M
	Operation and maintenance	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

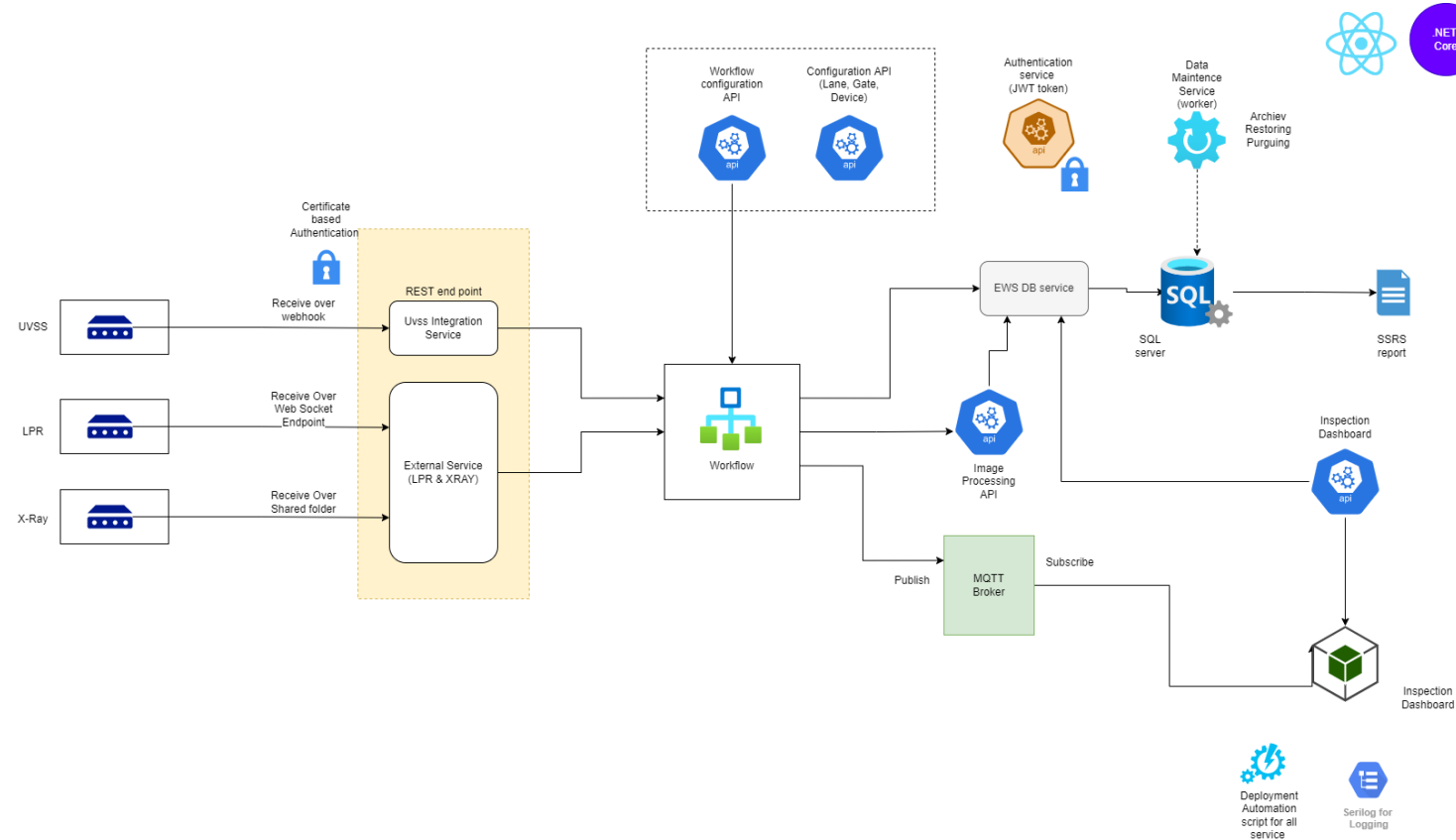
Software Architecture diagram – Context diagram



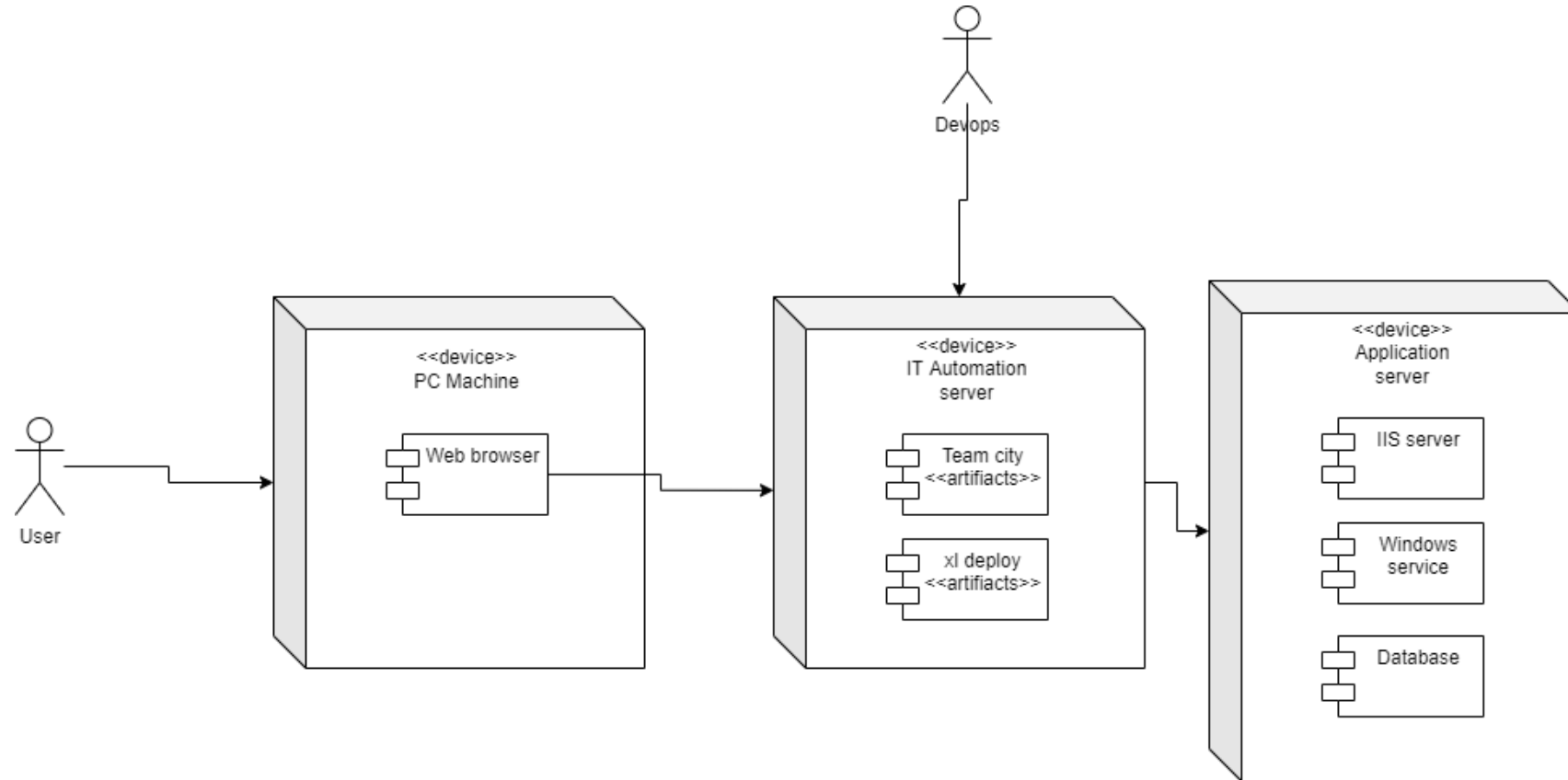
Software Architecture diagram – Module decomposition



Software Architecture diagram – Component & Connection diagram



Software Architecture diagram – Deployment diagram



How the system works

GM interface with third party systems like UVSS, LPR, X-RAY and provides a solution to control the entry of vehicle in site. System is designed to be product which can be configured and reuse for any site

- ❖ System is designed in Microservice based architecture, each service is based on
- ❖ Gate, Lane and device is configured in system based on any site. For eg., Lane is configured to Gate, and Gate to Devices (which is camera linked with LPR, UVSS or XRAY)
- ❖ After proper configuration, when vehicle crosses first gate, LPR device will send live vehicle detail to system. Same with UVSS and XRAY data
- ❖ Operator will check in system and vehicle physical documents

How the system works (Contd.)

- ❖ If everything is fine then Operator can approve and reject vehicle. If Operator is approved then vehicle will move to next gate or vehicle have to return back
- ❖ Created common workflow engine, which we can configured any API, GRPC call or MQTT events. This is core of system and can be reuse in any project
- ❖ System is capable of storing data for last 3 month, and get details on vehicle for any further reference
- ❖ Live data can be seen in system

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