CS 685 Spring 2017

Internet Of Things – Car's Ecosystem

Milestone 4

Car Fleet Management System

Team 2

Anthony Picaro – arp223@njit.edu

Nainika Aleti – na368@njit.edu

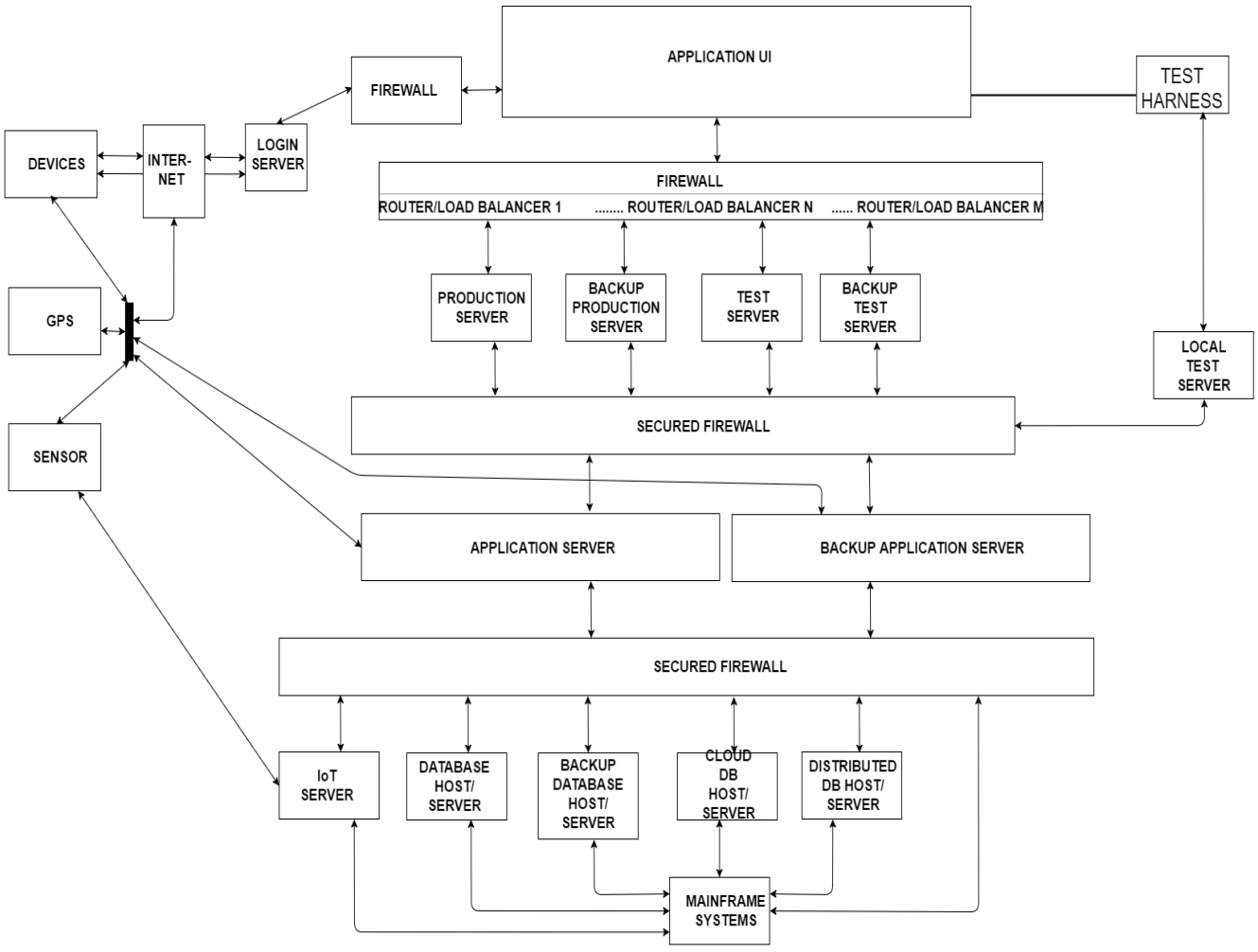
Vivek Bharathwaj Ravikumar – vr329@njit.edu

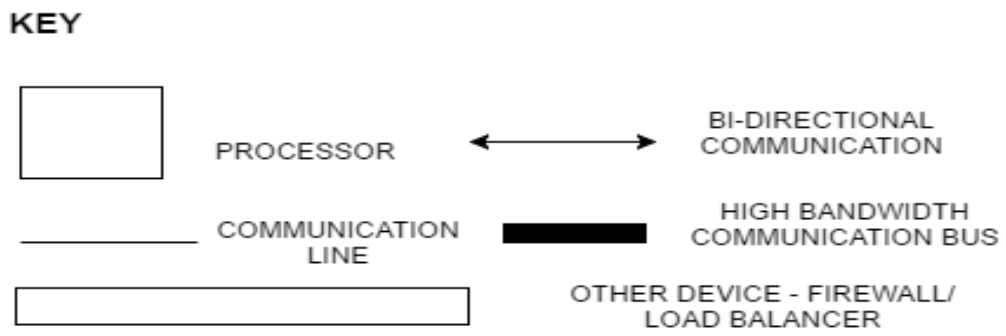
1. Introduction

In the fourth milestone, the creation of the Physical View diagram showing nodes and Connections became the focus of the project. The first task was to draw the Physical View Node DIagram; the rest of the milestone depends on the layers, so this was the main priority. Once this was done, the underlying driver and logic is explained for chosen style and pattern. Upon completion of the style and pattern descriptions, we were then able to show process that were created in milestone 2 and add them to a separate copy of the Physical View drawing, called as Physical View Process Diagram along with describing why each process are allocated to particular nodes.

The modules that were created in milestone 3 then deployed into the Nodes in the Physical view Architecture using a table. So relation between the M3 modules and the Process Architecture had to be established and explained. Lastly, but most importantly, the Physical View is useless if it has no bearing to the Reference Architecture, so their relation had to be explained, including all assumptions made about the Reference Architecture in order for the Physical View to work as intended, along with service requirements or assumptions.

2. Physical View Node Diagram





2.1 Why These Nodes?

The physical view explains about the non-functional requirements of the system such as availability, reliability, performance and scalability of a system is achieved. Also the executable environment of the software on a network of computers, nodes. Several different physical configuration has to be used for development purpose and testing purpose as the Test harness would be bit different from the Production environment.

APPLICATION UI – An app is as good as it's User Interface, this UI sits in the top node above all services and nodes as user access an application using an User Interface. The UI for a Test Environment would be bit different form that of the actual application as the application would still be under development under different phases. This Application User Interface satisfies the Non- Functional Requirement of **Availability**

LOGIN SERVER - The Login Server typically contains the User login and registration details required to login and access the application. This is provided and stored in the IoT User Registration Details.

DEVICES – The User access the app using devices like mobile and computers by accessing the Internet. Since the application runs on different mobile, web and desktop platform, any device can be used to access the app.

GPS – The GPS module is the physical Global Positioning System chip implemented into each user's smartphone that contains their real-time location. The GPS device is accessed by the user either while book and scheduling a car, for navigation and routing to destination and also tracking the position in real time. The Application servers also periodically access the GPS for same purpose. All the communication with GPS is connected with High Bandwidth Communication Bus, since GPS requires high speed and consumes high data.

SENSOR – The sensor reads and monitor the vehicle performance and has a data repository where it stores and maintains the data, for ex: if the fuel level goes down, the sensor stores the data into it. When this component is queried it sends notifications to the User interface component. This component also updates the IoT DB Server frequently since in a connected car environment, several cars could face the same issues which could be helpful in monitoring them

FIREWALL – A Secured Firewall is present in between different processing nodes in order to provide security at several layers, so that only authenticated users can access the underlying business logic and process. This firewall also acts a gateway between servers and servers and servers and database.

ROUTER/ LOAD BALANCER – A separate node beneath the Firewall is provided for the Router and Load Balancer. The Router transmits the packet from the application, like a web app, mobile app, desktop app etc. to its appropriate servers (web server, mobile server and desktop server). And since these packets are huge in

number, a load balancer used. This device that acts as a reverse proxy and distributes packets/ application traffic across a number of servers. They are used to increase capacity (**concurrency**) and **reliability** of application.

PRODUCTION AND BACKUP SERVER – The computing resources in the Production tier are used to run a Web Server, Mobile Server, Desktop Server (for example, Apache, Netscape, IIS, BEA WebLogic Express). The Entire application is tested thoroughly after several phases and loaded into a production environment. There should be no application code deployed to this tier. Only static should be served from this tier. Client requests for the application are forwarded to the Application tier via a proxy installed in the web server. Since there might be multiple production and test server to increase the scalability and concurrency.

TEST SERVER: These are the servers implemented during the testing phase of any update that may be released for the application. These servers house data that simulate a user's data, but does not actually contain any information related to any specific user.

APPLICATION AND BACKUP SERVER – The Application/ Backup Server hosts the Business process and logical components such as the functional components of a fleet like Vehicle Telematics Tracking, Fleet and driver monitor, Fuel and Speed Monitor, Vehicle Usage Analysis based on the information provided by Iot Server and the Sensor. This contains the code to implement these functionalities.

IoT Server - Iot Server refers to the Internet Of Things Server, an entity that has the entire information about all the users and cars stored electronically. They also provide several services in the bottom layer which can be accessed by the application in the top layer. For Ex, IoT database, this database can be accessed by the fleet application for Car and User Information.

Database/ Backup Host or Server - This Database is related and affiliated with the Fleet Application that store/ contains Information relevant and pertaining to the fleet application. This database in turn periodically updates the IoT DB Server. This has a backup which is frequently updated by the primary DB, so that in case of any catastrophic event, should the database to be damaged, this backup DB is triggered by the Fail-Safe plan. It acts as temporary DB and backup's the new DB

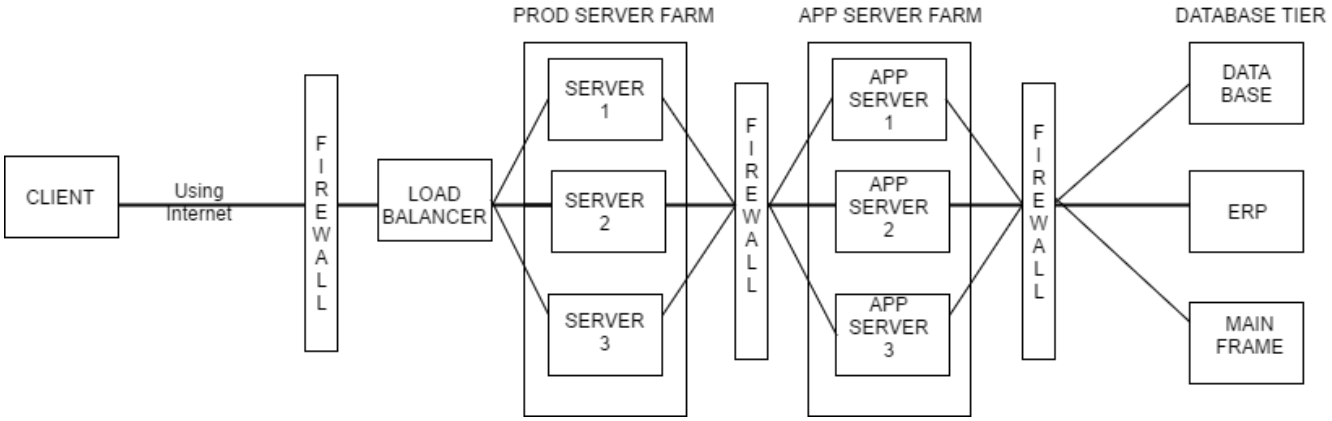
Cloud Database Host/Server – All the Fleet Related Information and data are stored in a remote, virtual environment call as cloud, which can be accesses from any computer easily. This Cloud server helps achieve the scalability and makes the database highly available.

Distributed Host/Server - A distributed database can avoid excessive traffic that is caused by the identical data that can be accessed locally. It appears as a single DB system, wherein it is a collection multiple connected databases that are spread physically across the globe and communicated using the Fleet's Network. This distributed server reduces the complexity in running a code.

Mainframe System – Having all the database is of no use unless they can be made into business. Mainframe systems can access these databases for bulk data processing for Enterprise Resource Planning, Transaction Management, SAP etc.

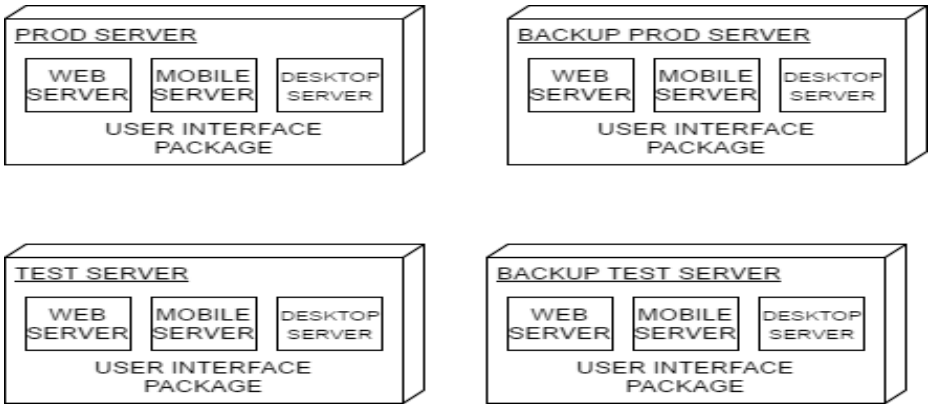
2.2 Styles And Pattern

2.2.1. Hardware Deployment Pattern



Client Server Physical Architecture - In this Style/ Pattern, the Client access the application using the internet. The Firewall blocks unauthorized user and the Load Balancer manages the network traffic. As the name suggests the Production server farm has several servers In order to satisfy the multiple user requests at the same time, this has the running code of the application's front end. These servers access the application server which hosts the code for the functional components of the system. The database tier is at the bottom, It has DB, ERP and Mainframe which are accessed by the Top layers.

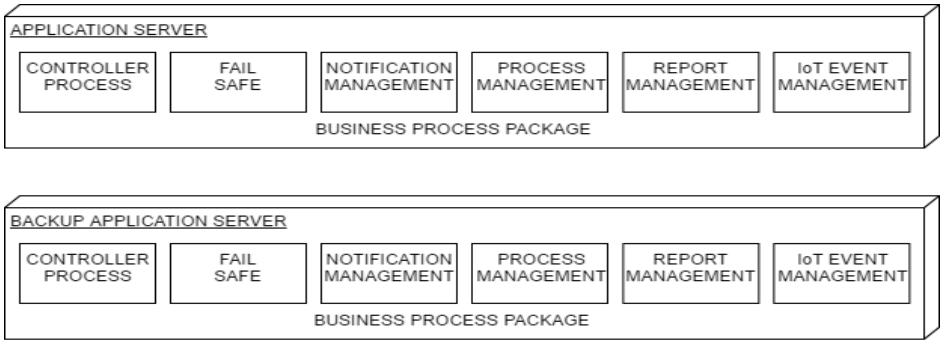
2.2.2. Production Server/ Test Server Deployment Pattern



Deployment Of Application into Server - All the Web, Mobile and Desktop app's UI's are loaded into their respective servers and packaged together and deployed into Prod/ Backup Prod server, Test/ Backup Test Server. The Test/ backup test server is used solely for testing purpose, they are deployed under several different environments to perform several testing techniques.

2.2.3. Application Server Deployment Pattern

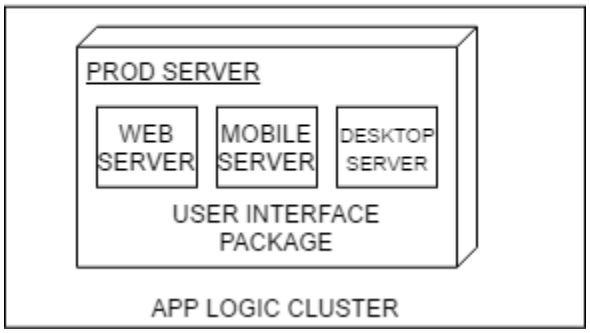
All the Business Process and Business Related Logic components are packaged and deployed into the Application and Backup Application Server. They are like Brain and Central Nervous System of the Application. They provide the logical answers to all the process. The Users who books a cab uses the process provided by this server, all the functional components are developed as code and deployed here.



2.2.3 Application Server Deployment

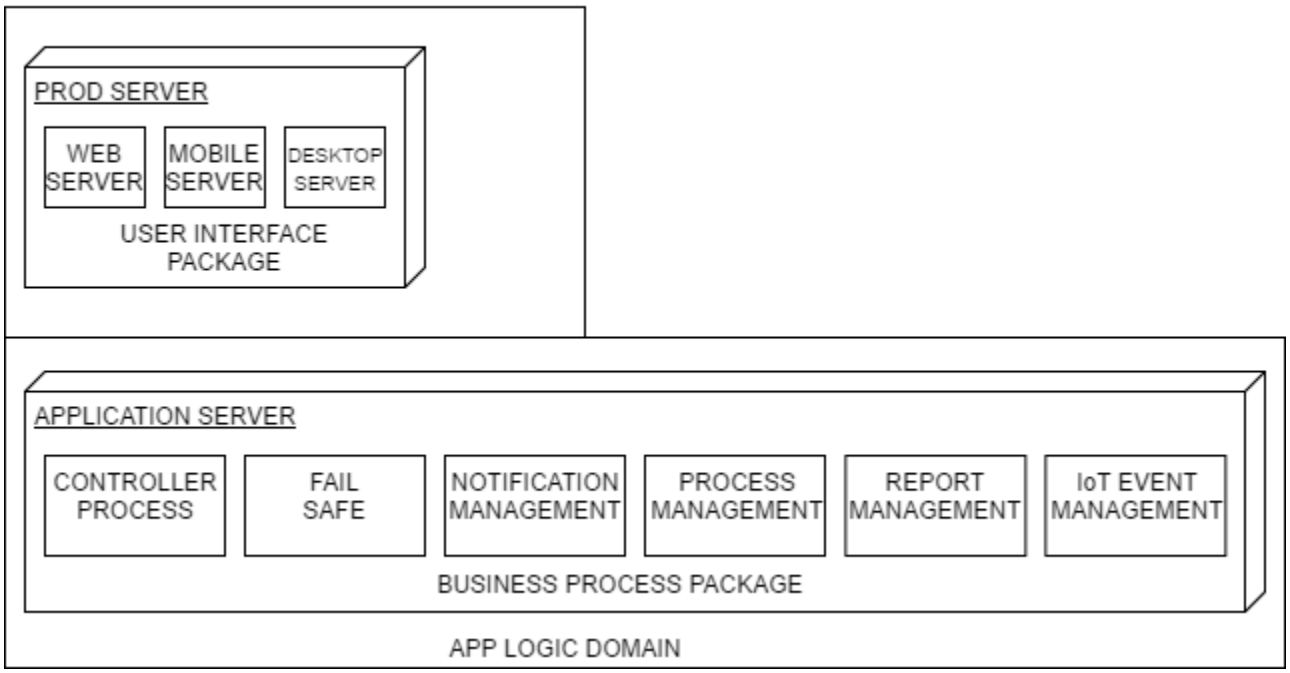
2.2.4. Domain Deployment Pattern

1. APP LOGIC CLUSTER



The First Diagram shows a single, clustered application deployed to a server in the Prod Server. This is one of the way to deploy an application

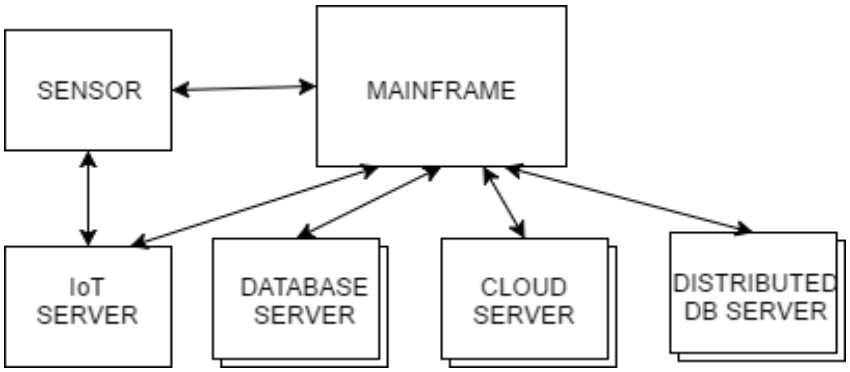
2. APP LOGIC DOMAIN



2.2.4.2 Domain Deployment – DOMAIN

The second diagram shows the deployment of a prod server on an application server. Such a deployment might be appropriate if the application is critical. The Prod Environment hosts the code required for the User Interface to Interact with process logic present in the application server. A single Prod server forms a App Logic Cluster, whereas the prod server and application server together forms a App logic Domain.

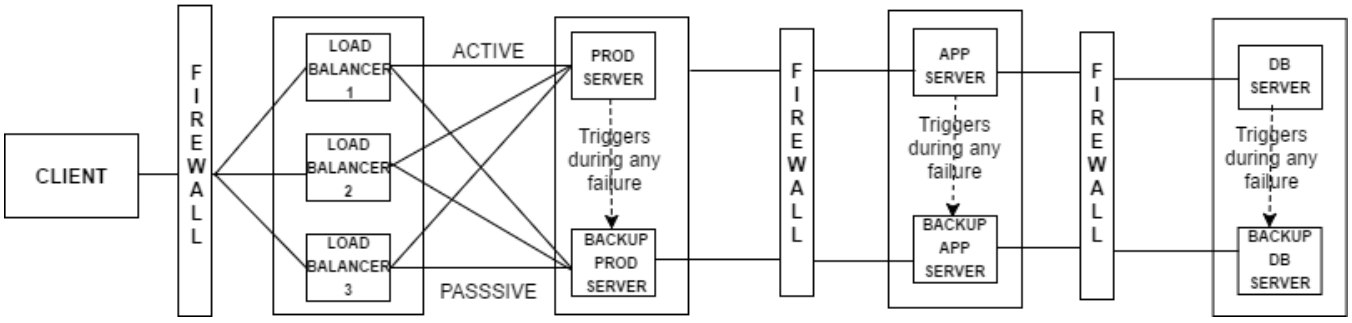
2.2.5. Database Deployment Pattern



2.2.5 Database Deployment

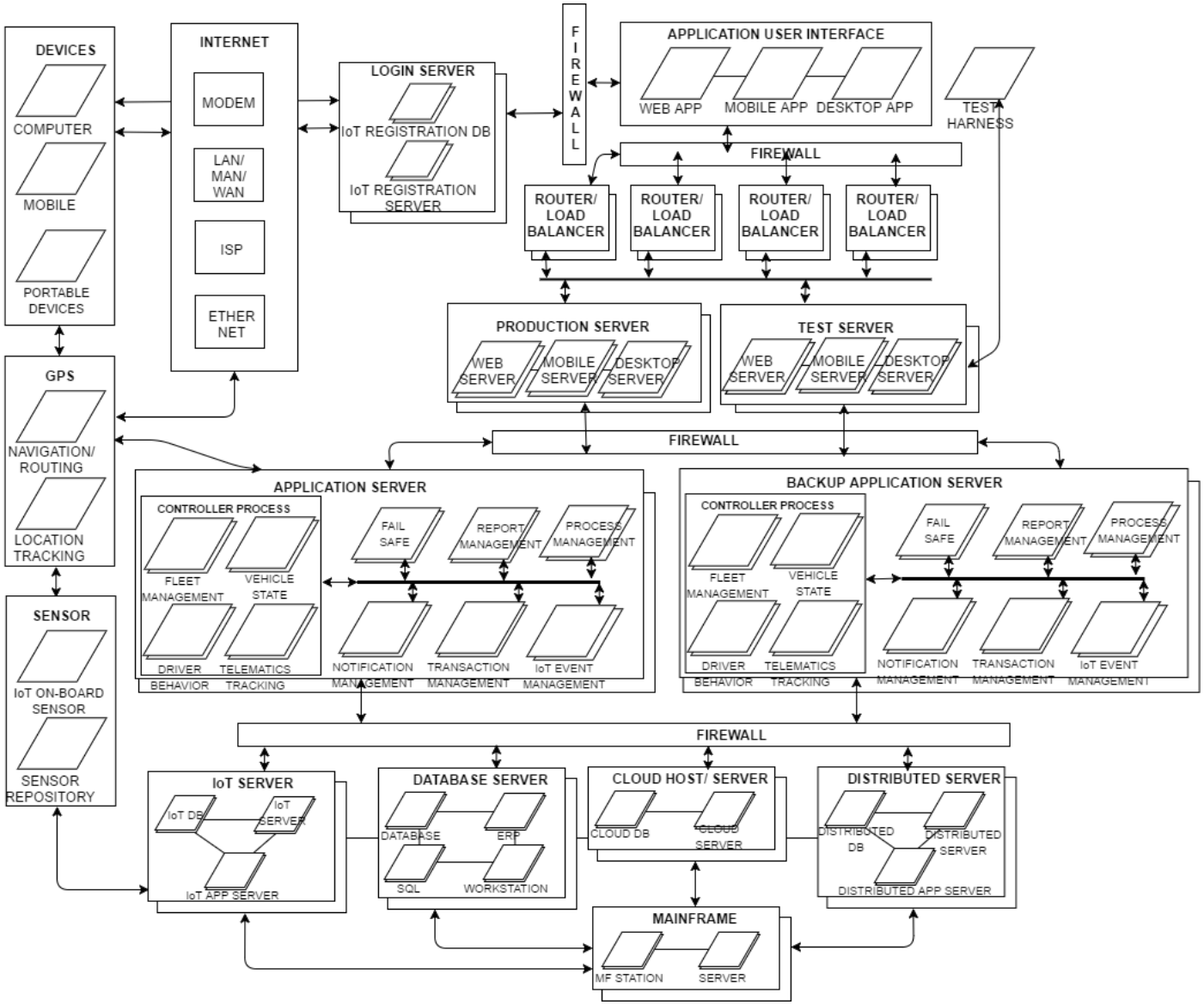
This Database Deployment pattern is chosen so that all these multiple and different database servers can be administered using a single entity, Mainframe. The Database tier is the primary location for storage of all business-related data and information for the business domain. Today, this data can be stored in an enterprise database management system, by a legacy or mainframe system. The Sensor Provides Information from the Vehicle and stores it onto the IoT Database server, which could be accessed through the Mainframe system. They provide vital information regarding the Car's health, Fuel level, Oil level, Telematics status etc.

2.2.6. Fail-Safe/ Load Balancer Pattern

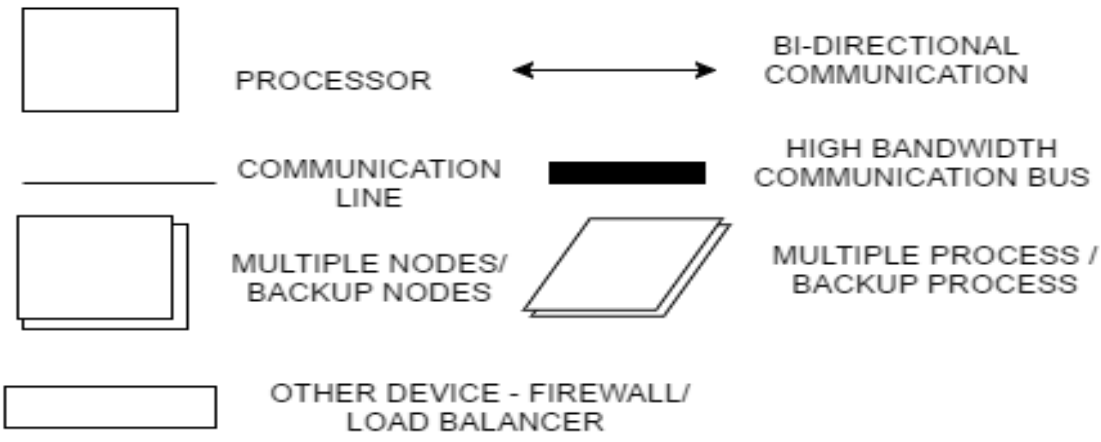


A fail-safe Pattern is a set of servers that are configured in such a way that if one server becomes unavailable, another server automatically takes over for the failed server and continues processing. The above diagram shows a failover pattern. It also shows a set of Load Balancer's which distributes the traffic evenly across the network.

3. Physical View Process Diagram



KEY



3.1 Allocation Of Process to Nodes.

3.1.1 Application User Interface Node

Web Browser: The Web Browser is one of the portals that any given user can access the application from, consisting of Firefox, Safari, Google Chrome, Internet Explorer, and Microsoft Edge. Web Browsers are multiplatform, so multiple kinds of devices can use the same browser, requiring each device's type to be taken into consideration when accessing the application.

Mobile App: The Mobile Application is the main method for a smartphone or a tablet to access the application, with specialized applications for Android, iOS, and Windows Mobile devices.

Desktop App: The Desktop Application is the preferred method to access the fleet management services and is the main method for computers and laptops, with versions for Windows, Mac OS, and Linux.

Test Harness: The Test Harness is how the fleet management software will be tested, consisting of a test environment and the application software version that is being tested.

3.1.2 Firewall/ Load Balancer Node

The firewall is the portion of the physical view that handles the security of the application, employing network policies, an application gateway, packet filtering, and an authentication mechanism to prevent malicious code/packets from reaching the servers. The Router/Load Balancer will reduce the load on any given router by splitting the workload equally among all of the routers to prevent bottlenecking, ensuring the best-possible connection to the software servers.

3.1.3 Production Server/ Test Server and Backup Node

The computing resources in the Production tier are used to run a Web Server, Mobile Server, Desktop Server (for example, Apache, Netscape, IIS, BEA WebLogic Express). There should be no application code deployed to this tier. Only static content (that is, images and HTML) should be served from this tier. Client requests for the application are forwarded to the Application tier via a proxy installed in the web server. Since there might be multiple production and test server to increase the scalability and concurrency.

These servers are also represented using two parallelograms one behind the other. Since the application needs to be tested under similar run time environments, these components are also deployed in the Test Server, so that while testing too, these components are available and testing can be done at 100% efficiency.

3.1.4 Application Server and Backup Node

The Application tier is where the application is deployed. The layers from the Development view in Milestone 3 is deployed in this tier. If high availability is desired, then a App cluster should be configured as part of the deployment process. This diagram shows the existence of a domain. The hardware could host one or many domains, and each domain could have one or many applications.

Conversely, if an application is deployed as a single file, then local interfaces and pass by reference can be used in calls between components in the application.

3.1.5 Database Systems Node

The Database tier is the primary location for storage of all important and business data and information for the business domain. Today, these data can be stored in an enterprise database management system, by a legacy or mainframe system and can be accessed for future purpose and also by using a ERP tool for Data Mining or Data warehousing, this vast amount of data can be used for prediction.

The Sensor Provides Information from the Vehicle and stores it onto the IoT Database server, which could be accessed through the Mainframe system. They provide vital information regarding the Car's health, Fuel level, Oil level, Telematics status etc.

IoT Server is an entity that has the entire information about all the users and cars stored electronically. They also provide several services in the bottom layer which can be accessed by the application in the top layer. For Ex, IoT database, this database can be accessed by the fleet application for Car and User Information.

All the Fleet Related Information and data are stored in a safe and reliable remote, virtual environment call as cloud, which can be accesses from any computer easily. This Cloud server helps achieve the scalability and makes the database highly available. A distributed database can avoid excessive traffic that is caused by the identical data that can be accessed locally. It appears as a single DB system, wherein it is a collection multiple connected databases that are spread physically across the globe and communicated using the Fleet's Network.

This distributed server reduces the complexity in running a code. Having all the database is of no use unless they can be made into business. Mainframe systems can access these databases for bulk data processing for Enterprise Resource Planning, Transaction Management, SAP etc.

3.1.6 GPS and Sensor Node

Since the Fleet application requires either an inbuilt or an external GPS device in order to do process like scheduling, Navigate, Optimize Route, Tracking. This GPS uses the services provided by the Application server in order to perform the above-mentioned task. It also uses the service provided by the IoT Architecture.

The Sensor Management Plays a vital role in the Fleet Application. An On-board sensor is mounted throughout the entire fleet, these sensors silently acquire and transmit the data to both Fleet Database and also IoT Server. The data is stored in the database which can be accessed later, it can be modified or deleted easily.

4. Module Deployment

MODULES	APP UI	ROUTER/ LOAD BALANCER/ BACKUP	PRODUCTION SERVER/ BACKUP	TEST SERVER/ BACKUP	APPLICATION SERVER/ BACKUP	DATABASE SERVER/ BACKUP	DEVICES	SENSOR
USER INTERFACE	Web Application, Mobile Application, Desktop Application, Test Harness	—	—	—	—	—	—	—
NETWORK MANAGEMENT	—	Router, Backup Router, Load Balancer and Backup Load Balancer	—	—	—	—	—	—
TEST HARNESS	—	—	—	Test Server and Backup Test Server	—	—	—	—
APPLICATION	—	—	Multiple Production Server and Backup Server	—	—	—	—	—
DEVICES	—	—	—	—	—	—	Mobile Phone - Android, iOS, Windows Computer - Windows, Mac, Unix/ Linux	—
BUSINSS PROCESS COMPONENTS	—	—	—	—	Controller Process, Transaction Management, Fail Safe, Process Management, IoT Event Management, Notification Management, Report Manaement	—	—	—
DATABASE	—	—	—	—	—	Database Server, Backup Database Server, Cloud Storage, Distributed Server, Mainframe, IoT Ecosystem Server	—	—
SENSOR	—	—	—	—	—	—	—	IoT Onboard Sensor, Sensor Repository
SECURITY	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL	SECURED FIREWALL

After Mapping the Modules that were identified in Milestone 3 with the Nodes From Physical View, the tabular column looks like above

UI Module is MAPPED TO APP UI Node - The UI module identified in the Development view is deployed into different platforms in the process view. The Web based platforms are – Chrome, Firefox, Safari, IE/ Edge. The Mobile based platforms (Android, iOS, Windows) The Desktop based platforms (Windows, Mac Os, Linux/ Unix)

Network Management Module is MAPPED TO Router /Load Balancer Node- The Network components are of prime importance in traversing a packet from source to destination, reduce the traffic, bandwidth etc.. This is achieved by employing Internet, Firewall, Router and Load Balancer and functioning hardware.

Test Harness Module is MAPPED TO Test Server/ Backup Node – The Entire Application for testing purpose is deployed in the Test Environment

Application Module is mapped to Production Server/ Backup node - The computing resources required to run in the Production tier are deployed using Web Server, Mobile Server, Desktop Server

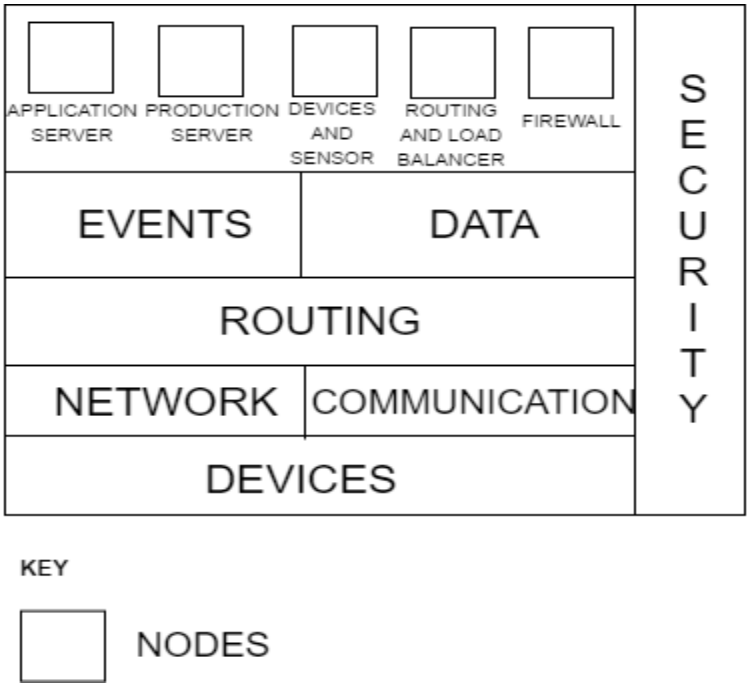
Device Module is Mapped to Device Node – Since the application needs to perform on multiple platforms across several devices, it is now deployed to be platform independent

Business Process Components Module is Mapped To Application Server/ Backup – All the Functional Components (ASR from M1) and other business components from development view are deployed.

Database Module is Mapped to Database Server Node- The Storage capacity for the Fleet Management System is now achieved by deploying all the database related servers in their respective nodes.

Security – Security Node provides security in the form of firewall between all the server and server and UI. This prevents any attack either on the system or on the database.

5. VIEW ALIGNMENT WITH REFERENCE ARCHITECTURE



Our Physical View fits into the application layer of the reference architecture and uses the services provided by the layers below it.

Application Server uses various network protocols to provide middleware services for security along with data access and persistence. They behave like an extended virtual machine for running applications, transparently handling connections to the database on one side, and, often, connections to the Web client on the other.

The production server sends client query directly to the database server and waits for a response. Once received, the web server formulates the response into an HTML file and sends it to your web browser. This back and forth communication between the server and database server happens every time a query is run. Contains Only Static code like html.

Database server access the data layer of the IoT reference architecture and performs tasks such as data analysis, storage, data manipulation, archiving, and other non-user specific tasks to provide database services to our fleet management application.

Devices and Sensors can securely access the applications database. They use the Iot Services of Devices and Events that are below them. Routing and Load Balancer – The router transmits the packets physically across the computer Networks using the Routing and Network layer in the reference architecture and the Load Balancer manages the traffic in the network. The Firewall node access the Security Layer provided by the IoT Reference Architecture to provide security to the application's architecture.

Reference:

1. Architectural Blueprints—"4 + 1" View Model of Software Architecture - Kruchten, Philippe. "
2. https://msdn.microsoft.com/en-us/library/ee658120.aspx