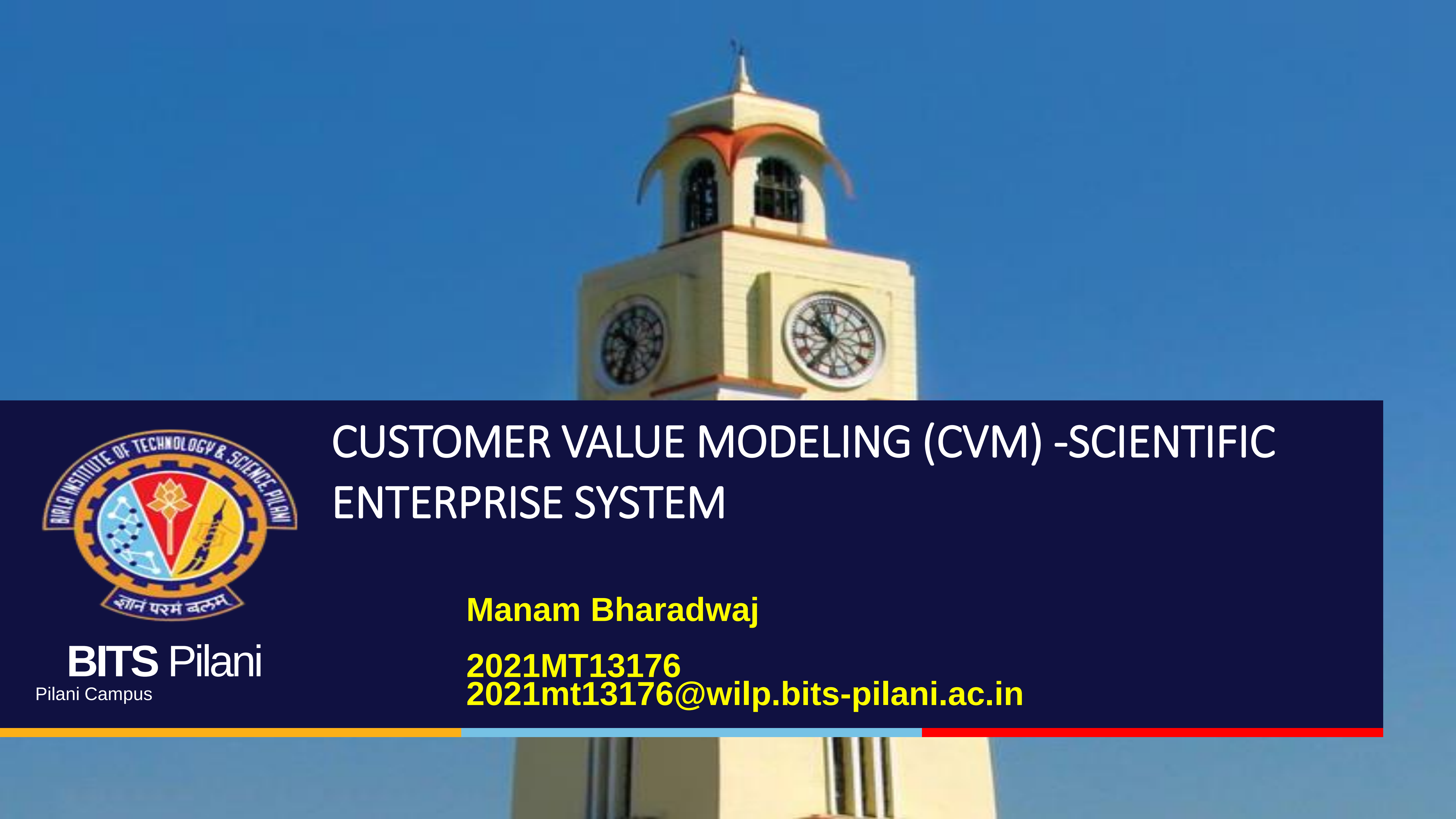


CUSTOMER VALUE MODELING (CVM) -SCIENTIFIC ENTERPRISE SYSTEM

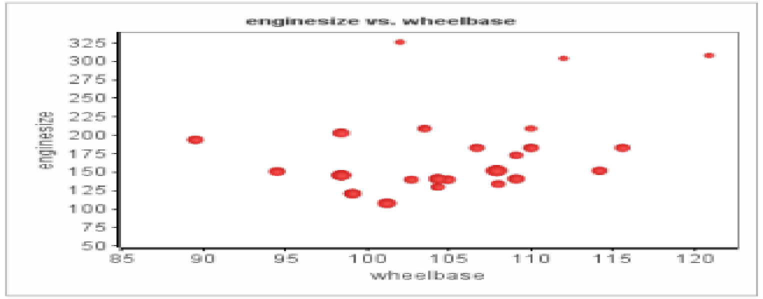
Manam Bharadwaj

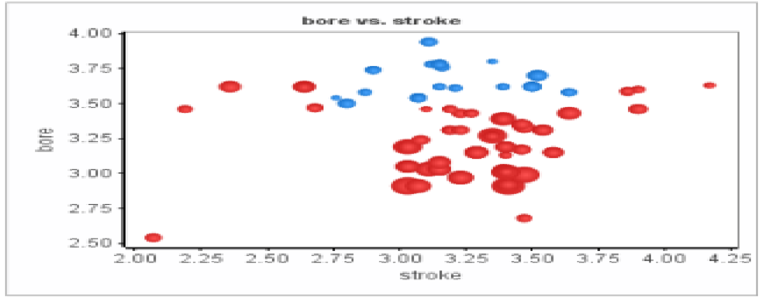
2021MT13176
2021mt13176@wilp.bits-pilani.ac.in

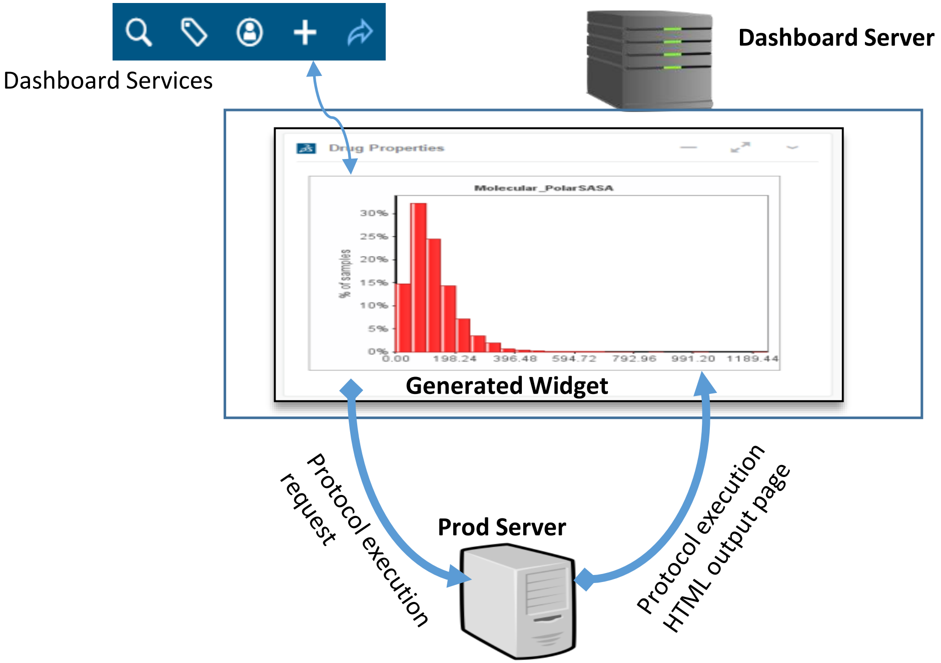
BITS Pilani
Pilani Campus

System and Goal

- This project is about my workspace where we have developed a Scientific Modeling System in the field of biodiversity and material science.
- The system works as robotic framework which leverage diverse data and then models it to reveal insights and flags related to future risks.
- Goal of the system is to have an highly efficient presentation system sitting on large unutilized data, processing it with easy interface to users and then presenting a single simplified view at real time.







Top3 Architecturally Significant Requirements (ASRs)

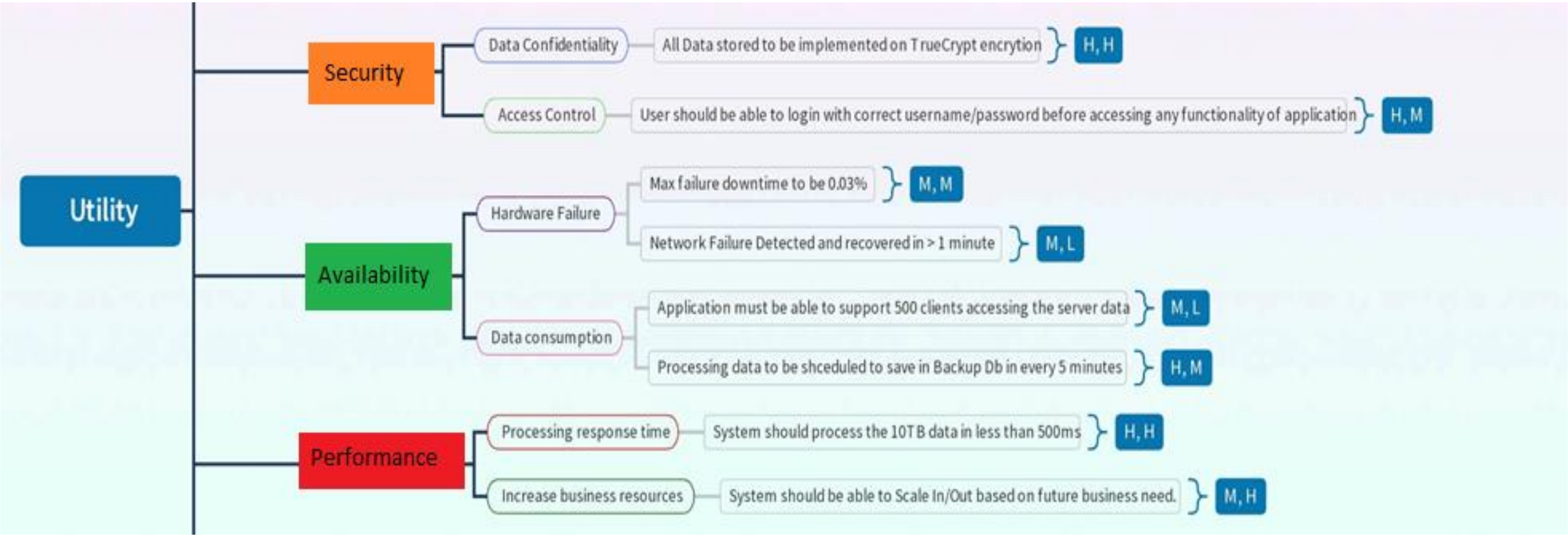
- **Availability** - The system should be available to customers at all times so that predictive analysis links are never broken, which may cause huge business loss to the enterprise science labs.

- **Security** - Data presented and in use is highly sensitive and users are required to be thoroughly validated against the SSO access services. This system may be used for strengthening or gathering information for publishing research as well. Any security breaches in such data would be liable to legal action which may tarnish brand value of the business along with financial penalty.

- **Performance** - Application must be able to display widgets to the dashboard with user request placed, and widgets should be ale to handle large streams inflow. We need to keep in mind that certain pattern may be particularly chosen to achieve this quality attribute hence this is important for the system.

Utility Tree Diagram

Importance, Difficulty to Implement



- Utility
  - Security
    - Data Confidentiality
      - All Data stored to be implemented on TrueCrypt encrytion – H, H
    - Access Control
      - User should be able to login with correct username/password before accessing any functionality of application – H, M
  - Availability
    - Hardware Failure
      - Max failure downtime to be 0.03% – M, M
      - Network Failure Detected and recovered in > 1 minute – M, L
    - Data consumption
      - Application must be able to support 500 clients accessing the server data – M, L
      - Processing data to be shceduled to save in Backup Db in every 5 minutes – H, M
  - Performance
    - Processing response time
      - System should process the 10TB data in less than 500ms – H, H
    - Increase business resources
      - System should be able to Scale In/Out based on future business need. – M, H

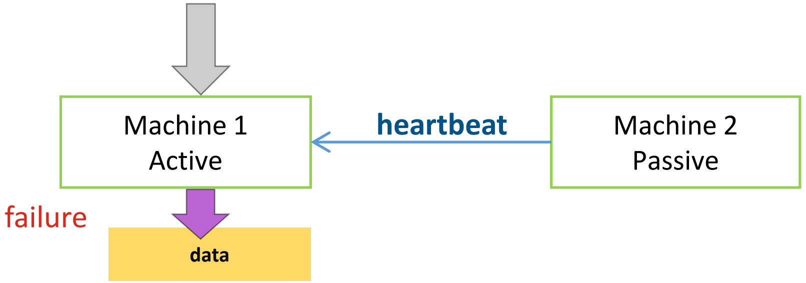
Use Cases

Use Case	Description
UC-1: Monitor Online Services	On-duty operations staff can monitor the current state of services and IT infrastructure (such as web server load, user activities, and errors) through a real-time operational dashboard, which enables them to quickly react to issues.
UC-2: Troubleshoot online service issues	Operations, support engineers, and developers can do troubleshooting and root-cause analysis on the latest collected logs by searching log patterns and filtering log messages.
UC-3: Provide management reports	Corporate users, such as IT and product managers, can see historical information through predefined (static) reports in a corporate BI (business intelligence) tool, such as those showing system load over time, product usage, service level agreement (SLA) violations, and quality of releases.
UC-4: Support data analytics	Data scientists and analysts can do ad hoc data analysis through SQL-like queries to find specific data patterns and correlations to improve infrastructure capacity planning and customer satisfaction
UC-5: Anomaly detection	The operations team should be notified 24/7 about any unusual behavior of the system. To support this notification plan, the system shall implement real-time anomaly detection and alerting (future requirement).
UC-6: Provide security reports	Security analysts should be provided with the ability to investigate potential security and compliance issues by exploring audit log entries that include destination and source addresses, a time stamp, and user login information (future requirement).

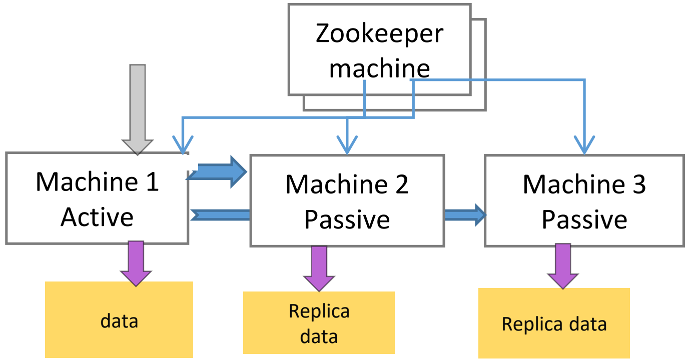
Tactics used to achieve Top 3 ASRs

Availability

- Server fault detection system at place using heartbeat mechanism



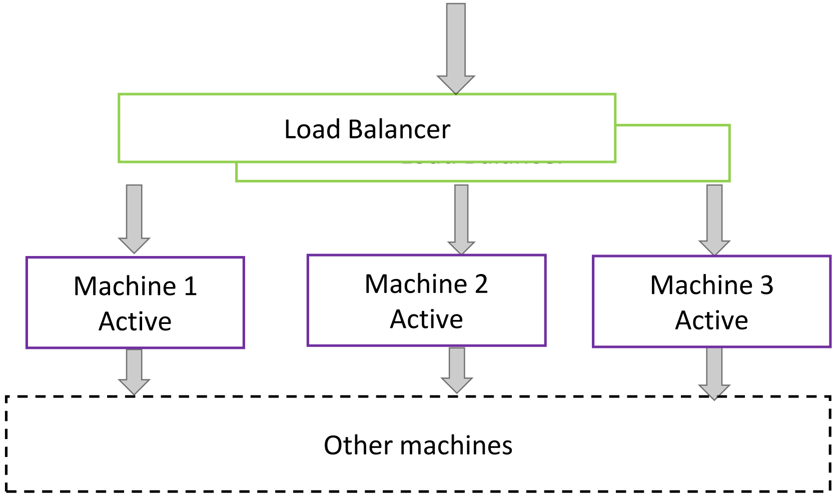
- Rollback and System Monitor at place for fault recovery and fault detection (Zookeeper)



Tactics used to achieve Top 3 ASRs

Security:

- Service Denial and Message detection at place via Load balancing systems
- Attack resistance is at place through data encryption and actor authorization with data access limit. This includes HMAC encryption technique
- System reboot and lock and availability check after restore is at place
- Ongoing and automated inspection and monitoring of web applets and enable Kerberos network security protocols

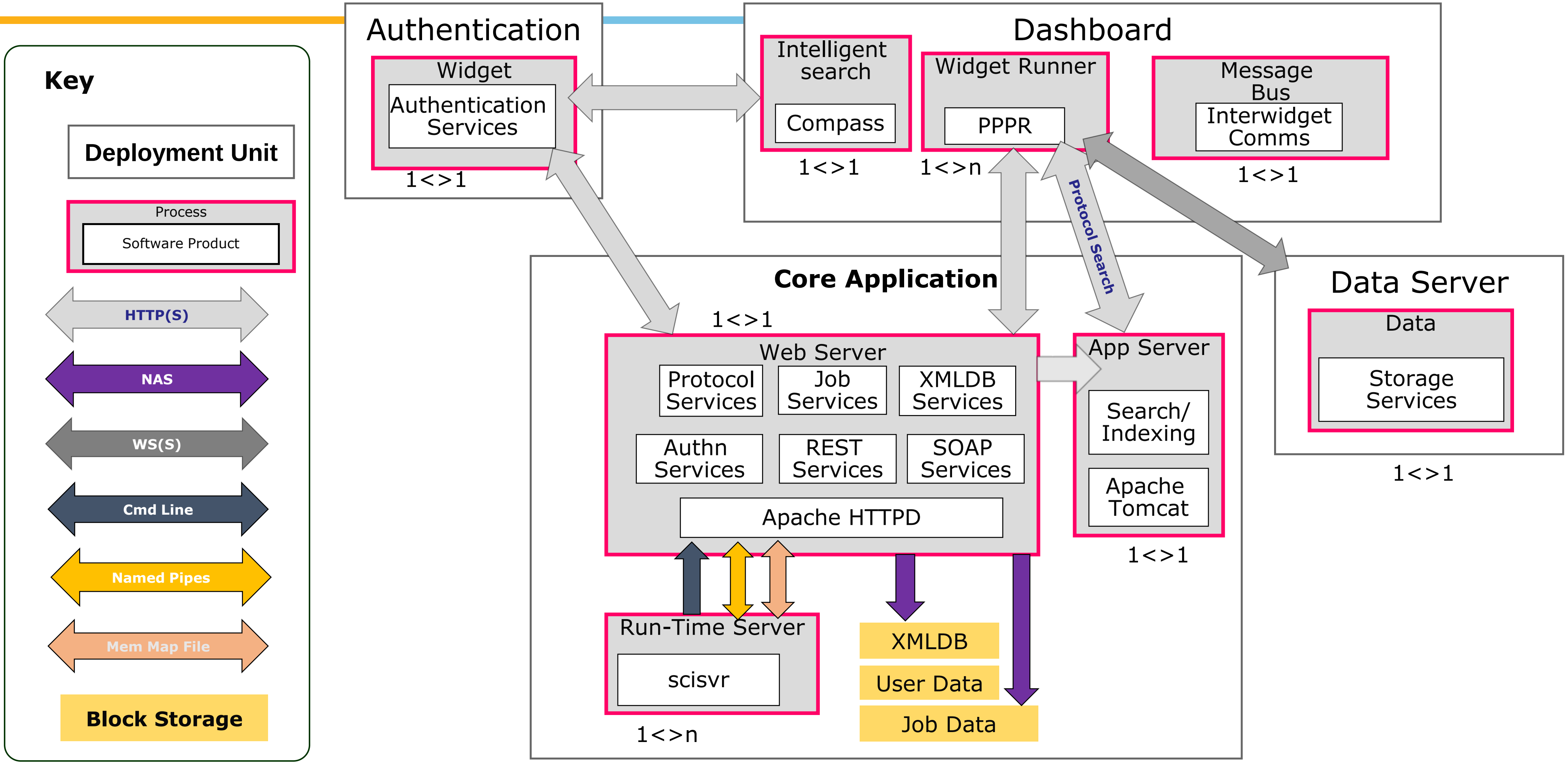


Tactics used to achieve Top 3 ASRs

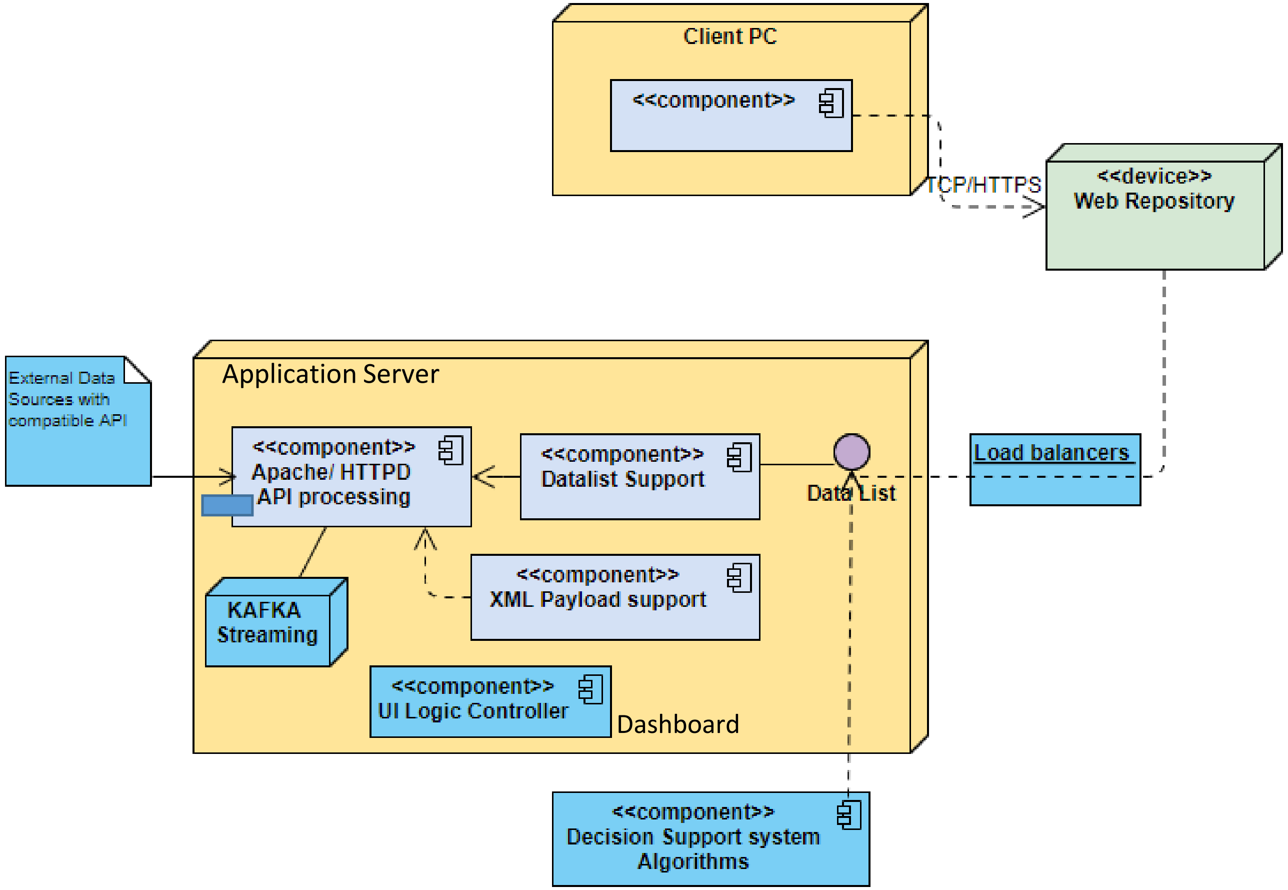
Performance:

- Resource demand and management handled with DDM at place for caching and faster retrieval to achieve maximum performance
- Extensive multithreading and background tasks are used for client application performance.
- Performance dashboard indicating multiple metrics set for real time monitor of performance
- Widget Runner and Widget Builder must not impose more than 2 seconds of overhead relative to execution of same protocol directly in browser.

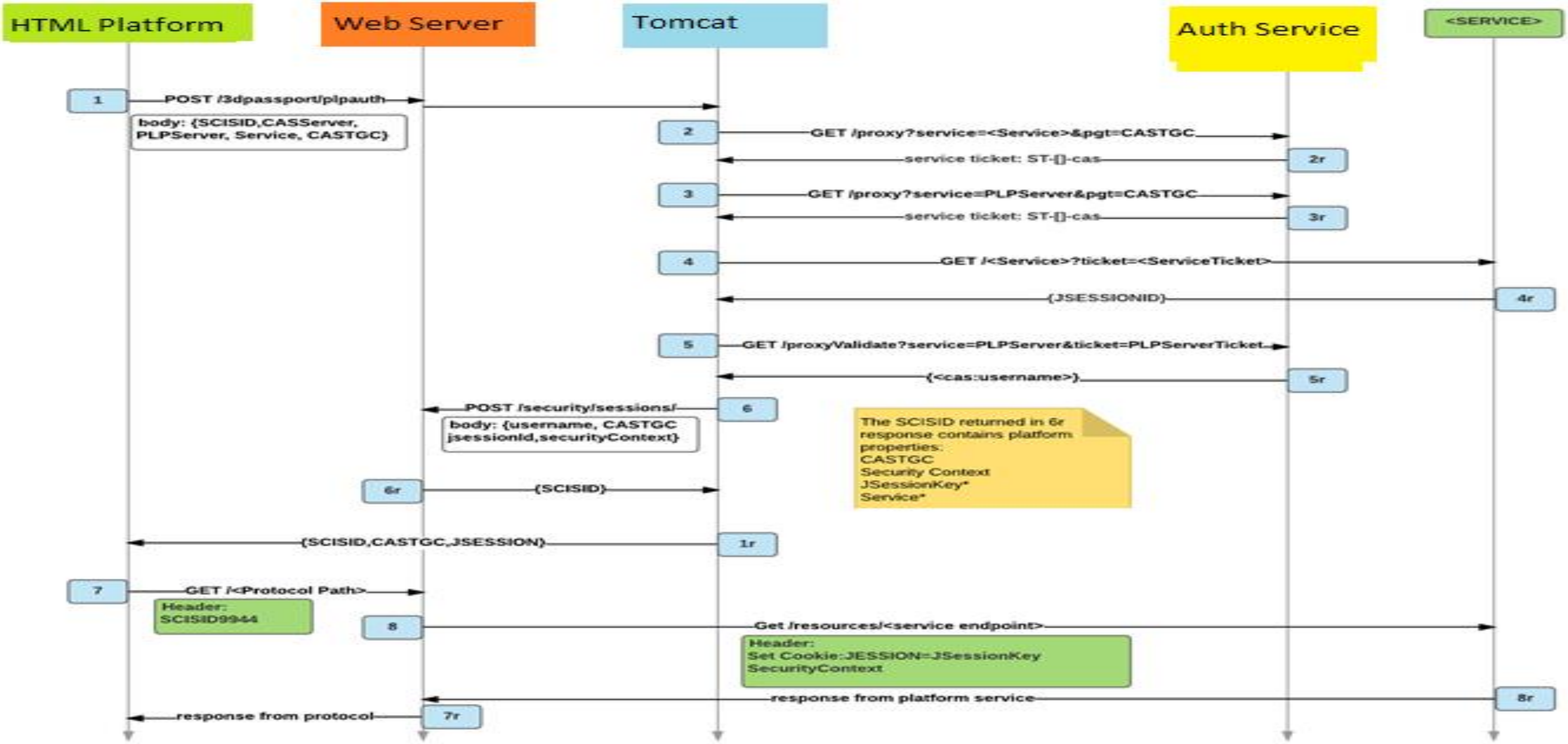
Component and Connection Diagram



Deployment Diagram



Sequence Diagram for Security Scenario: Access Control and authentication services



HTML Platform

Web Server

Tomcat

Auth Service

<SERVICE>

1 POST /3dpassport/plpauth

body: {SCISID,CASServer, PLPServer, Service, CASTGC}

2 GET /proxy?service=<Service>&pgt=CASTGC

2r service ticket: ST-[]-cas

3 GET /proxy?service=PLPServer&pgt=CASTGC

3r service ticket: ST-[]-cas

4 GET /<Service>?ticket=<ServiceTicket>

4r {JSESSIONID}

5 GET /proxyValidate?service=PLPServer&ticket=PLPServerTicket

5r {<cas:username>}

6 POST /security/sessions/

body: {username, CASTGC jsessionId,securityContext}

The SCISID returned in 6r response contains platform properties:
CASTGC
Security Context
JSessionKey*
Service*

6r {SCISID}

1r {SCISID,CASTGC,JSESSION}

7 GET /<Protocol Path>

Header:
SCISID9944

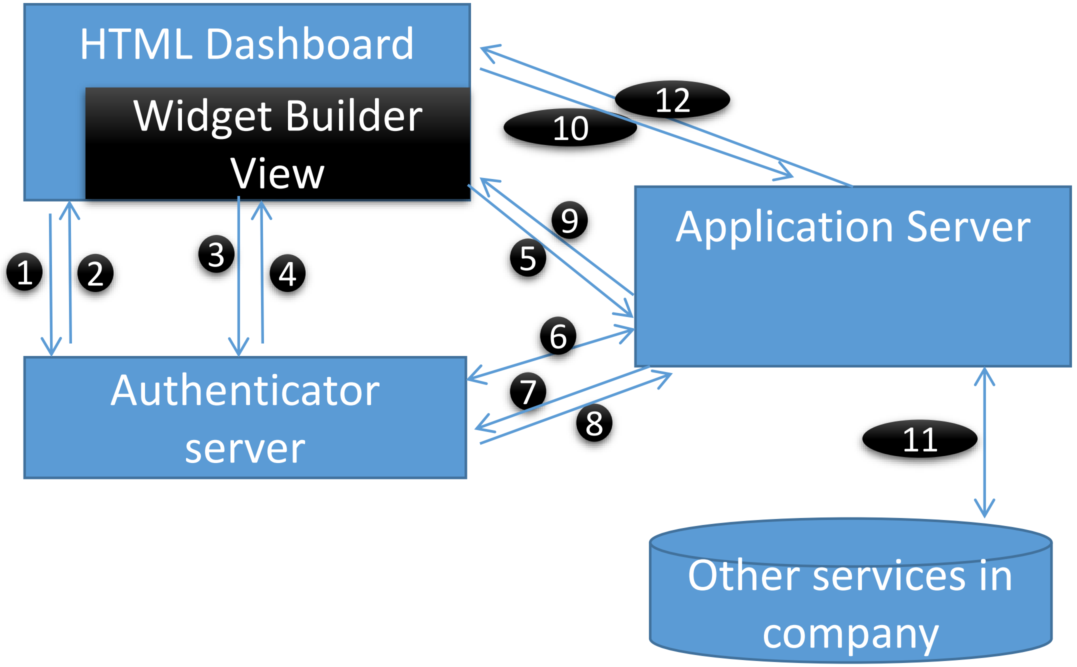
8 Get /resources/<service endpoint>

Header:
Set Cookie:JESSION=JSessionKey
SecurityContext

8r response from platform service

7r response from protocol

Security Scenario: Access Control and authentication services

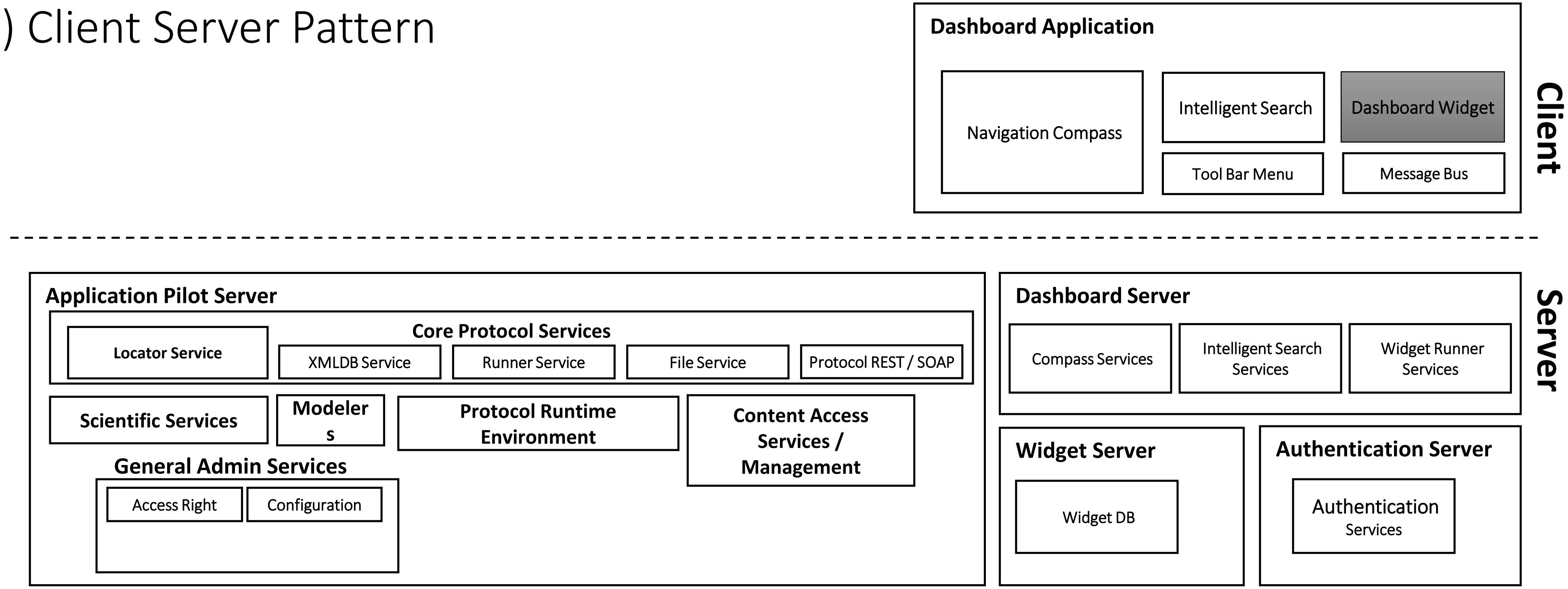


User can initiate the login and loading process which then triggers through Security and Authentication process respectively.

1. User tries Dashboard Login
2. Authenticator Server -Returns CASTGC cookie for session
3. Widget then Send a “CAS Login” request to the Authenticator Server.
4. Authenticator Return JSON Object, with Service Ticket Id
5. Widget then Send Authenticator Server URL and Service Ticket to the Application Server.
6. Application Server then Authenticate Service Ticket by checking with Authenticator messaging.
7. Get Services sessions request send to Authenticator.
8. Gets back Add returned JSessionIds to Application Session.
9. Application server then Return Session ID (SciSID) to Widget
10. Widget requests protocol execution (application running), passing through SciSID received in previous step.
11. Widget protocol exchanges data with other services using JSessionIds
12. **Result page returned to Widget Viewer**

Patterns Used:

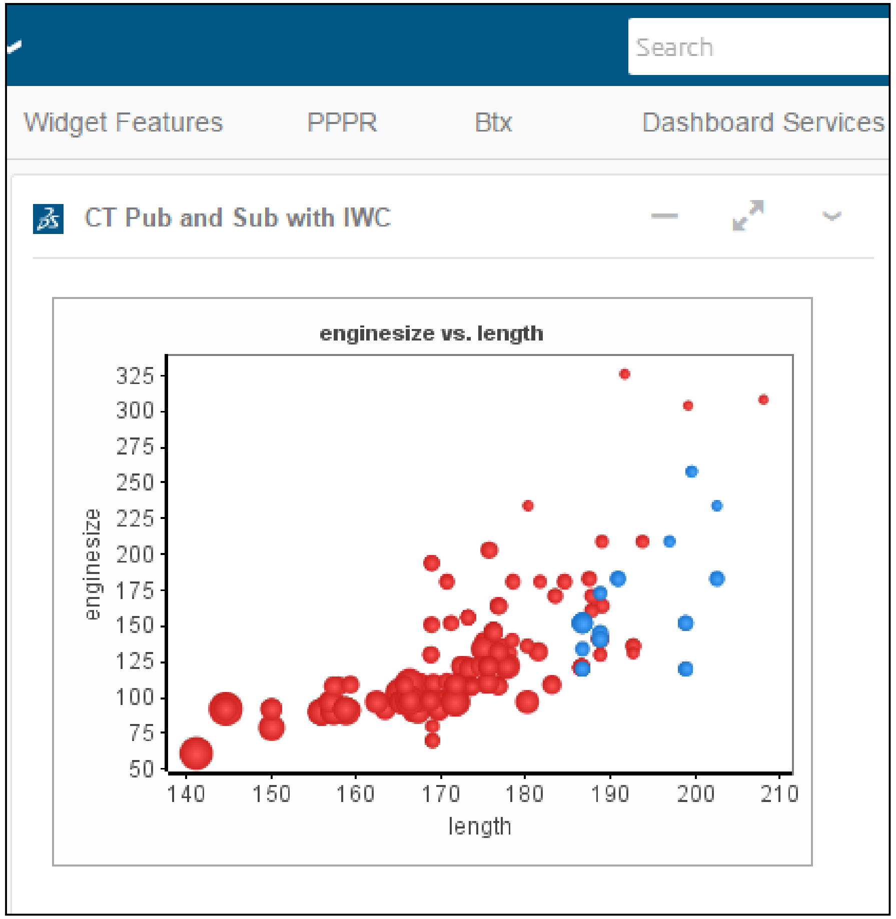
A) Client Server Pattern



- Used to provide data/management services to all stakeholders as described earlier in the views.
- For management operations CLI/GUI resides on client desktop/laptop systems and REST web server is on widgets.
- For Data operations, clients from distributed hosts would be requesting services from protocol server.
- Client calls are made on Bidirectional-Asynchronous Message communication pattern

Patterns Used:

B) Publish-Subscribe Pattern: Bidirectionality



- Widgets using ‘explicit tagging’ (Tagging where all tag info is passed to Tagger on Widget init server) can set UI elements connected to the Data Connector to ‘**Publish and Subscribe**’.

- In this case user selections will be converted into new Tags (Category: User Selections) named ‘Selection n’ which can then be used by the Tagger system like any other Tag.

<u>Actions</u>:

Create and view multiple Selections
Union with other User Selections
Intersect with other Tags

How System Works

- System is working as Client Server Architecture
- There are two type of Clients which communicate with a central server
- All sort of Client and Server communication happens through central server through TCP/IP.
- Data Visualization Stream happened with a socket implemented on server
- Both clients have implementation of Algorithm applets and Logic controller which can be shared through a central server.
- Communication between Client and Server happens with REST/SOAP services and Socket connection
- Server follows the Micro services Architecture to provide REST APIs and Socket Connections
- System connects to external data sources via an extensive library of APIs
- Creates and combines interactive charts and visuals for key stakeholders Fully HTML 5 compliant.
- Graphs include: **Scatter plot**, **Line Plots**, **Pie Charts**, **Bar Plots**, **Histograms**, **Polar Plots** and **Interactive tables with sorting**, **search**.

Key Learnings

1. **Learnt to handle following challenges in architectural planning and design phase.**
   a. What is the time available for the architectural analysis/evaluation? It is challenging enough to come up with one solution, let alone a few!
   b. What is the product pipeline for the next 1–3 years? And what other projects are lined up? Can we see any synergies?
   c. What is system current technical debt that we could potentially address?
2. **Learnt to handle following technical, high level and low level design aspects of architecture planning**
   a. Such system require must plan for ASRs and NFRs at architectural planning level.
   b. The SOLID principles just do not only apply on software development but also when architecting a system
   c. Client system is live for an hour at a time so error handlings (Network, Hardware issues) must be at its best and well designed
   d. Patterns involved in the system as different type of users using a complex UI.
   e. System availability and performance is most important concern and must be handled properly as system needs to be available throughout the year.
   f. Component connections can not be easily modifiable so must be planned before hand in architecture.
3. **Learnt to understand different architectural patterns used to fulfill requirement goals.**
   a. C&C, deployment diagrams, sequence diagrams are very useful to get clarity on the environment, elements and their interaction, visualize the flow or interdependencies and end result in an early stage.
   b. Architectural documentation helps new team members/stakeholders to quickly ramp-up and have confidence in the overall product.

THANK YOU