Storage Tracking, Allocation and Replication Tool (START)

long internship Project Submitted

In partial fulfilment of the requirements for the award of the degree

Of

BACHELOR OF TECHNOLOGY

By

P. Abhinaya

Roll No: R170754

Under the supervision of

B. LingaMurthy

(Assistant Professor)

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge Technologies, RK Valley

Idupulapaya, Kadapa(Dist), Andhra Pradesh

Rajiv Gandhi University of Knowledge Technologies,RK Valley

Idupulapaya, Kadapa (Dist), Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled “Storage Allocation, Tracking and Replication Tool (START)” is a long internship submitted by P. Abhinaya (R170754) in the department of Computer Science and Engineering in partial fulfilment of requirements for the award of degree of Bachelor of Technology for the year 2022-2023 carried out the work under the supervision

Internal Guide	HEAD OF THE DEPARTMENT
Mr. LINGAMURTHY	Mr. SATYANANDARAM
(Assistant Professor)	**(Assistant Professor)**

Project Coordinator
M.MUNI BABU
(Assistant Professor)

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-RKValley, Kadapa Dist - 516330

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, “Storage Allocation, Tracking and Replication Tool (START)” is the Bonafede work of P. Abhinaya (R170754). Here by accord our approval of it as a study carried out and presented in a manner required for its acceptance Major of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed, or conclusion drawn, as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

LingaMurthy
Project Supervisor
Dept. of CSE
RGUKT IIIT RKValley

Examiner
Project Examiner
Lecturer Dept. of CSE
RGUKT IIIT RKValley

RAJIV GANDHI UNIVERSITY OF KNOWLEDGETECHNOLOGIES
(A.P. Government Act 18 of 2008)
RGUKT-RKValley, Kadapa Dist – 516330

DECLARATION

I am **P. Abhinaya (R170754)** hereby declare that the project report entitled “Storage Allocation, Tracking and Replication Tool (START)” done under the guidance of **Mr LingaMurthy** is submitted for minor project of **Bachelor of Technology** in **Computer Science and Engineering,** is an authentic record of our own work carried out under the supervision of **P. Abhinaya,** the Major Project December 2022 - January 2023 at RGUKT – RK Valley.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references.

The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

P. Abhinaya (R170754)
Date: 20-01-2023
Place: RK Valley.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success. We are extremely grateful to our respected Director, **Prof. K. Sandhya Rani** Mam for fostering an excellent academic climate in our institution. We also express my sincere gratitude to our respected Head of the Department **Mr. Satyanandaram** Sir for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project. We would like to convey thanks to my project guide **Mr. LingaMurthy** Sir for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly for the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

CONTENTS

ABSTRACT

START (Storage Tracking, Allocation, and Replication Tool)

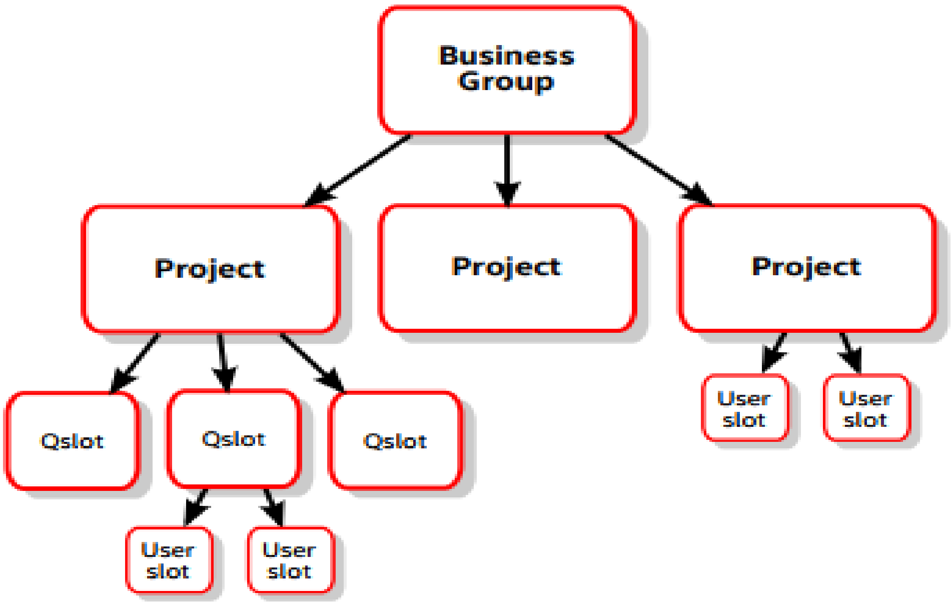
START (Storage Tracking, Allocation, and Replication Tool) is a global storage management system. START allows users to Create storage areas, delete storage areas, Change the size of existing areas, extend the lifespan of existing areas, create replicas of areas at remote sites, which can be automatically updated at regular intervals, Cache areas for use of multiple remote sites, View information about file servers, disks, and areas. Users can work with START through the command line or through the START Web Interface. The Procedure for creating, deleting, and resizing area on certain types of file server is manual i.e., it involves sending a request email or opening a GEMS ticket. This means that these operations cannot be completed instantly. So, there is a delay between the time when the request is made and the time when it is carried out.

OVERVIEW

START (Storage Tracking, Allocation, and Replication Tool)

START (**S**torage **T**racking, **A**llocation, and **R**eplication **T**ool) is a global storage management system which provides standardization and transparency of storage operations for different types of file servers within and across sites. START is developed to deal with the problem of no consistent storage management processes or tools with a vision “Storage ATM”. Customers ought to be able to provision and manage disks themselves, without bothering a storage admin, just as one can get cash from an ATM himself, without waiting for a bank teller. ATMs are only safe because banks have policies that detect problems and determine how much cash one can withdraw at a given point in time. Likewise, our ATM vision of data management requires tools to let storage admins easily define storage and replication management policies. START provides functionalities like Auto-discovery, Information collection, enhanced interface for viewing status of storage, history database, profiles, and group functionality. START's extensible architecture enables it to work with any type of fileserver. It is supplied with drivers that allow it to work with NetApp, EMC, CMODE fileservers. Additional drivers can be created to work other types of file servers. As a tester, I am responsible to perform validation for the features developed by the developers for the START either by manual or automation. START has a hierarchical Resource Allocation Structure consisting of

Business Groups, projects, and, optionally, Qslots and user slots. This structure is illustrated in the following diagram:



Business Group and project nodes correspond to actual Business Groups and projects. Storage resources are allocated to project nodes only. Qslots allow limits (for example, on area size or number of areas), defaults, and policies to be defined at the sub-project level. User slots allow them to be defined for specific users or for all users who do not have their own user slots. If a particular limit or attribute is not set for a certain project, Qslot, or user slot, it inherits the limit from its parent in the hierarchy. For instructions on how to configure Business Groups, projects, Qslots, and user slots, allocate resources to them, and define limits and defaults for their areas.

OTHER FUNCTIONALITIES OF START

Group functionality: START allows the administrator and certain designated users within a project to create and manage areas for other group members. Storage areas can belong to a individual user or to a group.

Auto-discovery: START has the ability to discover fileservers, disks, and storage areas, and to track and manage areas that were not created through it. This behavior is configurable.

Information collection: properties can be defined for file servers, disks, and areas. (For areas, you can specify optional and/or required properties that can/must be given values when an area is requested.) Plugins (scripts) can be written to collect this data.

Enhanced interface for viewing status of storage: START includes commands for viewing details of file servers, disks, areas, requests, the Resource Allocation Structure, groups, permissions, and file server drivers, including historical data

History database: START stores information about file servers, disks, areas, users, groups, and events in a database. This history data can be queried and analyzed to spot trends.

Profiles: you can set default parameters for area creation.

Roles

User

Describes the START user role. A user is someone who uses START to create and manage storage areas.

START Administrator

Describes the START Administrator role. The START Administrator is responsible for: • Allocating resources to Business Groups and projects based on their needs

• Configuring START to automatically discover file servers, disks, and existing areas. • Manually configuring START to work with file servers, disks, and areas where necessary.

• Configuring limits for the number of areas, sizes of individual areas, and area lifespans as required by Business Groups and projects.

• Writing plugins (scripts) to enable START to work with other types of file servers.

Group Administrator

Describes the group administrator role. A group administrator is a user who is typically responsible for a project's areas. The group administrator can create and manage areas for other members of the project and can also reactivate expired areas.

TOOLS & FRAMEWORKS

ION (Interactive Over Netbatch) :

ION - **I**nteractive **O**ver **N**etbatch - is an extension of Netbatch for managing interactive server resources. ION Sessions Manager is developed by Intel and is used by 1 user of Software Informer ION provides a new intended use model which allows better dynamic allocation of datacentre resources, improved reservation/isolation of interactive resources, and greater ease of use and flexibility for resource allocators. In Unix computing, **Ion** is a tiling and tabbing window manager for the X window system. It is designed such that it is possible to manage windows using only a keyboard, without needing a mouse. In computing, a **tiling window manager** is a window manager with an organization of the screen into mutually non-overlapping frames, as opposed to the more common approach (used by stacking window managers) of coordinate-based stacking of overlapping objects (windows) that tries to fully emulate the desktop metaphor.

VNC (Virtual network computing):

VNC session is created in ion session manager, VNC session is used through ion session manager. VNC viewer and VNC server are designed to work best together, use them both to enhanced security, more reliable connections, and access to exclusive features. VNC is a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer. It transmits the keyboard and mouse input from one computer to another, relaying the graphical-screen updates, over a network. VNC is platform-independent – there are clients and servers for many GUI-based operating systems and for Java. Multiple clients may connect to a VNC server at the same time. Popular uses for this technology include remote technical support and accessing files on one's work computer from one's home computer, or vice versa

IntelliJ IDE:

It is an integrated development environment (IDE) written in Java for developing computer software written in Java, Kotlin, Groovy, and other JVM -based languages. It is developed by JetBrains and is available as an Apache 2 Licensed community edition. IntelliJ IDEA integrates the essential developer tools and lets you debug, analyze, and version the code base of your applications from within the IDE. The IDE provides certain features like code completion by analysing the context, code navigation which allows jumping to a class or declaration in the code directly, code refactoring, code debugging , linting and options to fix inconsistencies via suggestions. IntelliJ supports plugins through which one can add additional functionality to the IDE

Jenkins:

Jenkins is a popular open-source automation server that is used for continuous integration, continuous delivery, and continuous deployment (CI/CD) of software projects. It provides a web-based user interface that enables developers to create, schedule and monitor automated jobs, known as “builds”. It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands. Plugins have been released for Jenkins that extend its use to projects written in languages than Java. Plugins are available for integrating Jenkins with most version control systems and bug databases. Many build tools are supported via their respective plugins. Plugins can also change the way Jenkins looks or add new functionality.

Gitlab:

GitLab is a web-based git-based repository management tool that provides a complete DevOps platform for managing the entire software development life cycle. It offers a variety of features, including source code management the management continuous integration and delivery (CI/CD), Issue tracking project management. It is a complete DevOps platform that enables professionals to perform all the tasks in a project—from project planning and source code management to monitoring and security. GitLab enables portfolio planning and management through epics, groups (programs) and milestones to organize and track progress. Regardless of your methodology from Waterfall to DevSecOps, GitLab's simple and flexible approach to planning meets the needs of small teams to large enterprises.

Youtrack:

Youtrack is a web based issue tracking and project management tool developed by JetBrains. It is designed to help development teams collaborate and manage their projects efficiently. Key features of Youtrack includes Issue Tracking, Agile Project Management, collaboration tools, customization, Integration, Security. YouTrack is a tool to manage your projects, configure agile boards, keep knowledge base, use reports and Gantt charts, dashboards and time tracking. Track tasks and issues, support customers' requests, plan sprints and releases, create workflows, and never force your
process to fit the limits of a tool again! YouTrack speaks your language, supports markdown and emoji-reactions. YouTrack is flexible enough to follow your exact process

JUNIT Framework:

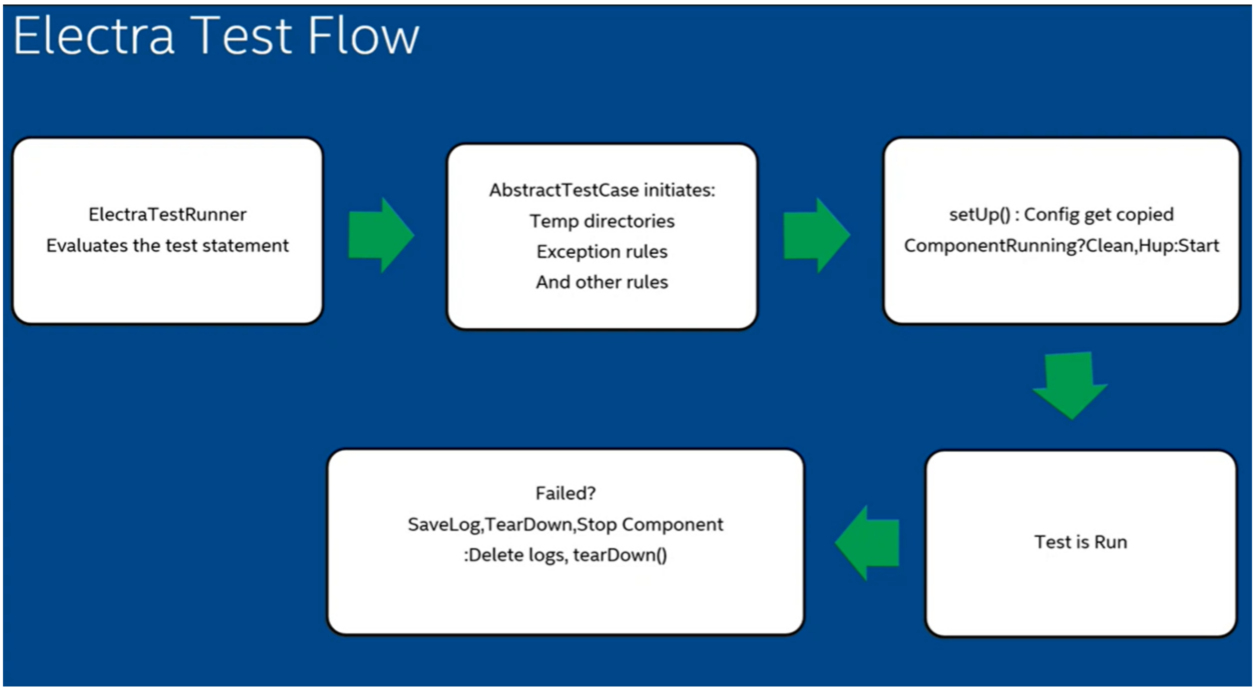
Junit is a popular open-source testing framework for java based applications, It provides a set of annotations and APIs for writing and running unit tests for your java code. JUNIT is widely used by developers and software testing professionals for its simplicity and ease of use. This framework allows developers to write automated tests for their code, making it easier to catch errors in the daily life cycle. It provides wide range of features, including test fixtures, assertions and test runners, which make it easy to create and run tests. JUnit is the most famous framework for writing unit tests in Java. You write test methods that call the actual methods to be tested. The approach is like "test a little, code a little, test a little, code a little…”. It increases programmer productivity and stability of program code The test case verifies the behaviour of the code by asserting the return value against the expected value, given the parameters passed.

START GUI:

This is the gui where all the START related tasks in the production are displayed we can check whether the submitted jobs and tasks are running correctly or not. It enables users to create areas, caches, replications, and get informed about their deletion period. Users can track their requests from different sites in START web console. Users are allowed to see log files if there is any cause of failure. Developers can find the operations that users are performing on the web console.

Electra Framework:

Electra is an end-to-end testing framework which can test all layers of a software (backend, web service, frontend etc.). It has the capability to test a software using interfaces like command line interface (CLI), UI and http. It is written in JAVA and provides strong and user-friendly API. Electra is **Junit** based Framework. Junit is a unit testing framework for the Java programming language. It is important in the test-driven development. This approach is like “test a little, code a little, test a little, code a little…”. It increases programmer productivity and stability of program code. Electra is not just for functional testing, we use Electra for performance testing, regression testing, exceptional testing. Using Electra, we can validate multiple products by just adding application Installation path in the configuration file.



Electra Test Flow

ElectraTestRunner
Evaluates the test statement

AbstractTestCase initiates:
Temp directories
Exception rules
And other rules

setUp() : Config get copied
ComponentRunning?Clean,Hup:Start

Test is Run

Failed?
SaveLog,TearDown,Stop Component
:Delete logs, tearDown()

Electra Test Start-up Sequence

1. Electra configuration file -> ~/.electra/application
2. Copy test specific configuration
3. Replace patterns in configuration files name and contents

   USER

   _PROJECT_ROOT_

   _TEST_ROOT_

   _TEST_CONF_

   SANDBOX

4. Check if the needed components are running from the needed path and needed version

   If no,

   “force clean” components (persistency)

   Start components

   If yes,

   “clean” components

   Hup components

Electra Test Finish Sequence

1. Clean components
2. Check exceptions
3. If test failed
   - Stop all components
   - “force clean”
   - Save
     - test log
     - components logs/startup files
     - configuration on each error

Electra Specific Features

- Adding site specific properties
- Annotation specific for regression

- Changing configuration while running test
- Configuration checks
- Error reporting
- Override TestFlow class
- Mailing
- Emulators

START ARCHITECTURE

The START server:

• Accepts area creation, replication, and management requests, status requests, and administrative requests. It handles them as follows:

• Local area creation and management requests--invokes the appropriate plugin (from the appropriate driver) to carry out the request on the file server. For some requests, it does not need to invoke a plugin. Instead, it simply makes changes to its own internal representation of the area. (For example, to change an area's lifespan, it does not need to do anything to the area itself.)

• Remote area creation and management requests--sends the request to the appropriate remote START server.

• Area replication requests:

1. Invokes the appropriate plugin to request that the area be created on the file server (local or remote).

2. Sends an area creation request to the remote START server(s).

3. Creates a replication task.

When the time comes to run a replication task (according to the interval and time specified by the user), the START server sends the replication plugin (script) to its dedicated Netbatch Pool1 (composed of existing rsync servers). The script is run on one of the rsync servers, which uses rsync to copy the data from the original area/directory to the replica area.

- Status requests--retrieves the requested information from the database and returns the results to the user.

- Administrative requests--performs the requested action or returns the requested information.

- Records events (area creation, etc.) and information about file servers, disks, areas, users, and groups in the history database.

The Project Configuration Service:

- Maintains a list of permitted Intel projects and filesystem types

- Can be used by multiple START servers.

- Can be configured to run on multiple machines to provide failover capabilities

REQUEST LIFECYCLE

Describes the stages that a request (for example, an area creation request) goes through. When a user makes a request (for example, for an area to be created), it goes through several stages.

These stages are as follows:

Initializing: The request is being initialized. (Requests do not remain in this stage for very long.)

Waiting For Approval: If the request is one that requires approval, it remains in this state until the person responsible for approving requests) approves it. (The START Administrator can specify that requests that meet certain criteria, for example, creation of an area greater than 1GB in size, require approval.)

Frozen: If the request is for an area to be deleted and a Legal event hold notice applies to the area, it can be frozen until the end of the hold period.

In progress: START has requested that the action be carried out. For some types of file server, the request can be carried out immediately. For others, the request procedure may include sending an email request or opening a GEMS ticket, which can take time.

Success: The request was completed successfully

Failure: The request could not be completed

Cancelled: The request was cancelled.

TESTING PROCESS

Every feature and bug are created in form of issues in Youtrack. Youtrack tracks tasks and issues, support customers’ requests, plan sprints and releases, create workflows. Each issue travels through several stages before deployment. These stages include open, working, code review, testing, pending release, and done. When a feature is brought to testing state, testers are supposed to write testcases based on the use cases written by the developers. If required, a request discussion is conducted with developer to understand the functionalities of the feature and the testing procedure need to be followed (manual or automation). The testcases written are sent to review by the developers in which they suggest if anything else need to be validated. After getting approval for the testcases, the actual testing starts. Automation or manual tests are conducted based on the requirement. Whatever the testing procedure, the results or tests are need to be reviewed by the team lead and the automation tests are validated based on regression results. Jenkins is used to run regression for the automated tests. If the tests are inconsistently passing or failing consistently, the tests need to be checked again fi there is anything missing like adding site properties.

MANUAL TESTING

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not.Manual testing is performed when a feature or bug need to be tested in Integration. Integration involves dealing with actual fileservers, disks, and areas. Manual testing is generally carried out by running Linux commands on the real environment to check if the feature is working as expected or not. We should perform this with utmost care as it is dealing with real fileservers. The outputs are captured and sent for review. If anything, unexpected happens in this process, an open bug issue is raised and forwarded to developer for the fix. After the fix, bugs are again tested and feature itself is tested again to make sure it is working fine. And then the issue is moved to pending release state, corresponding testcases are moved to implemented state and the bug is moved to done state.

AUTOMATION TESTING

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation involves the similar process of manual testing but rather than testing the feature on real environment by running commands, automation is done in **Electra Framework**.

To perform automation, we majorly have three repositories

1. Common
2. Common.testing
3. Start.testing

Common: All the common framework exists here.

- common.testing/java/framework

Common.testing: It includes application specific tests written to validate the classes in common repo. Common and common.testing are needed to be cloned for every project.

- Common.testing/java/netbatch
- Common.testing/java/start
- Common.testing/java/nbflow

Start.testing: This repo includes all the tests that are written for the START framework.

- Start.testing/java/tests

REGRESSION TESTING

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to ensure that the product works fine with new functionality, bug fixes or any changes to the existing feature. After the completion of automation Nightly Branch regression is ran to ensure the successful working of feature or bug fix on production. Only the tests that passed regression are merged into master. Nightly Build regressions are run after deployment of every new build. As a tester we perform fixes for the failed tests in regression to make sure that new updates are not impacting the old features. Jenkins is the tool we use to perform any kind of regression.

USER MODEL TESTING

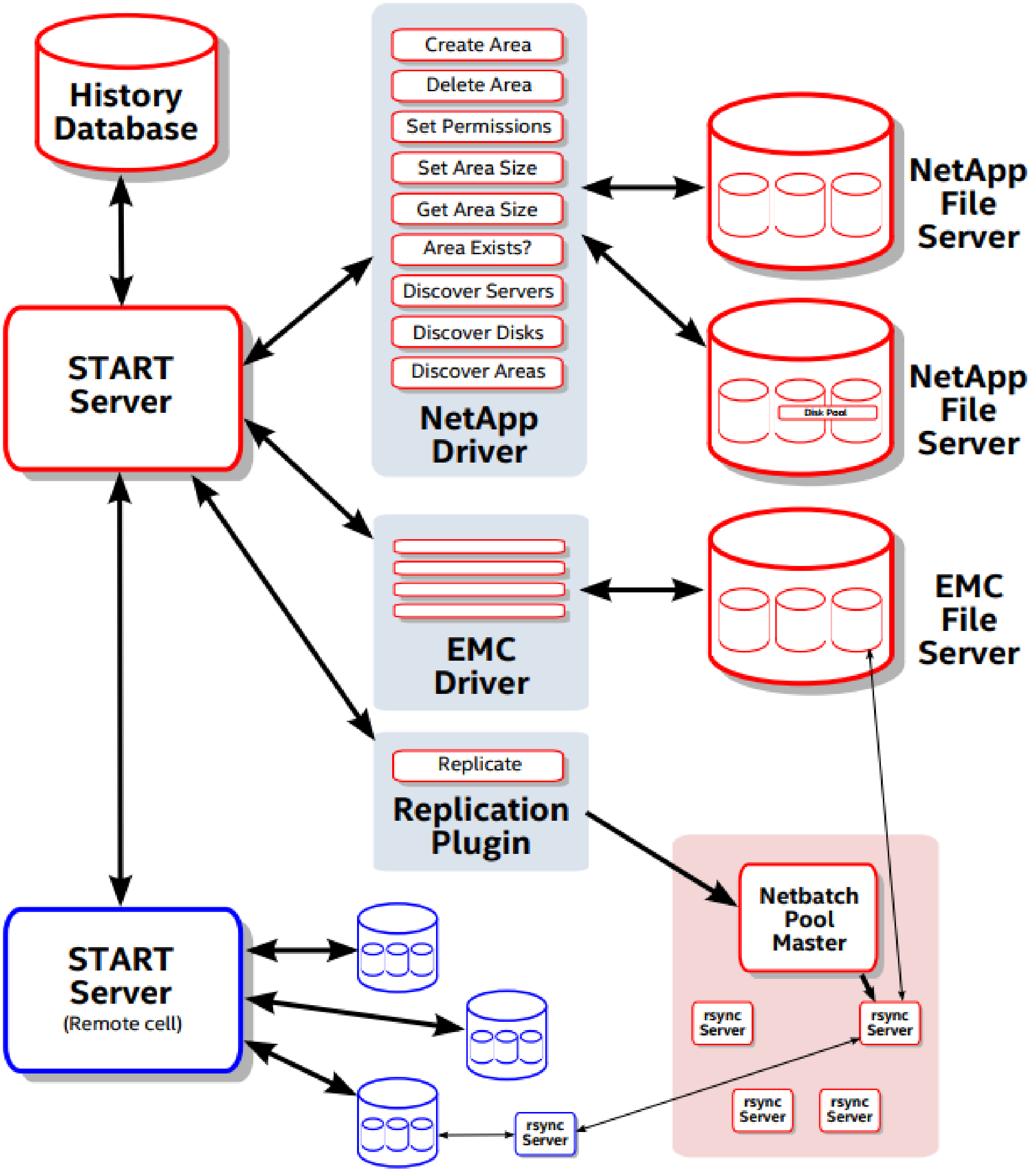
UMT includes fetching persistency from production servers like listing out areas, disks, fileservers present on each site and comparing the output with previous build results to make sure they are not affected.

TEST FLOW

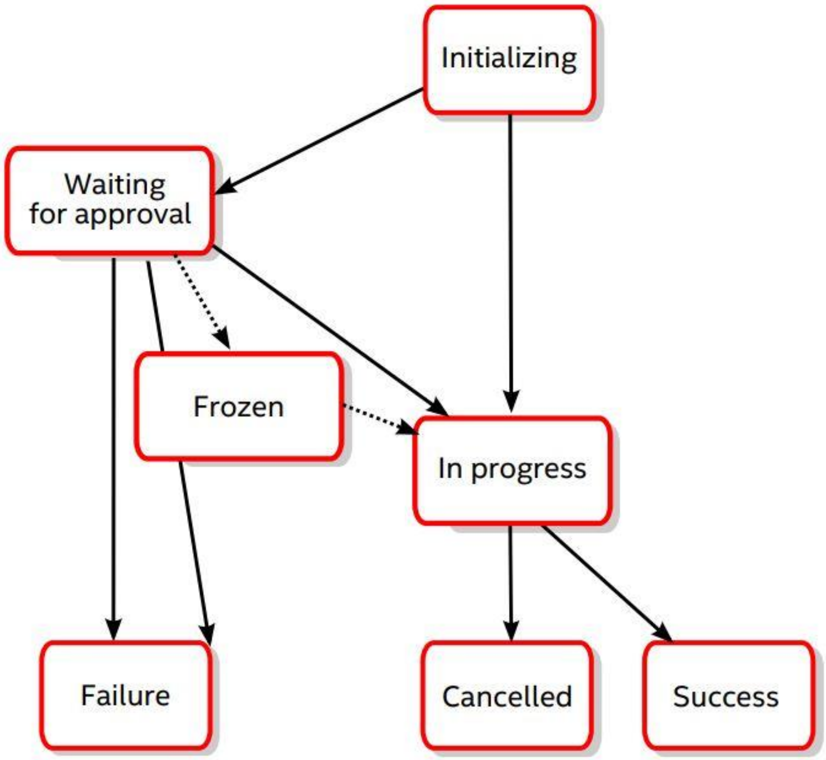
- The initial step in the test flow is the identification and understanding of the issue. Issue is nothing but a ticket (either a bug or feature ticket).
- Identify the type of the issue
- If it is a bug ticket
    - Understand the scenario
    - Write the testcases for the sequence
    - Get the testcases reviewed by the developer
    - Automate the tests
    - Reproduce the bug in the found in version
    - Check if the fix is working fine in the current version
- If it is a feature ticket
    - Understand the feature
    - Schedule a request discussion about the feature with the developer
    - Based on the findings, fill the impact matrix
    - Write down possible test cases
    - Get them reviewed from the developer
    - After the approval, automate the testcases in electra
    - Get the tests reviewed from the team lead
    - Run branch regression for the tests
    - Merge the branch after approval
    - Make sure that the tests are passing in regression

BLOCK DIAGRAMS

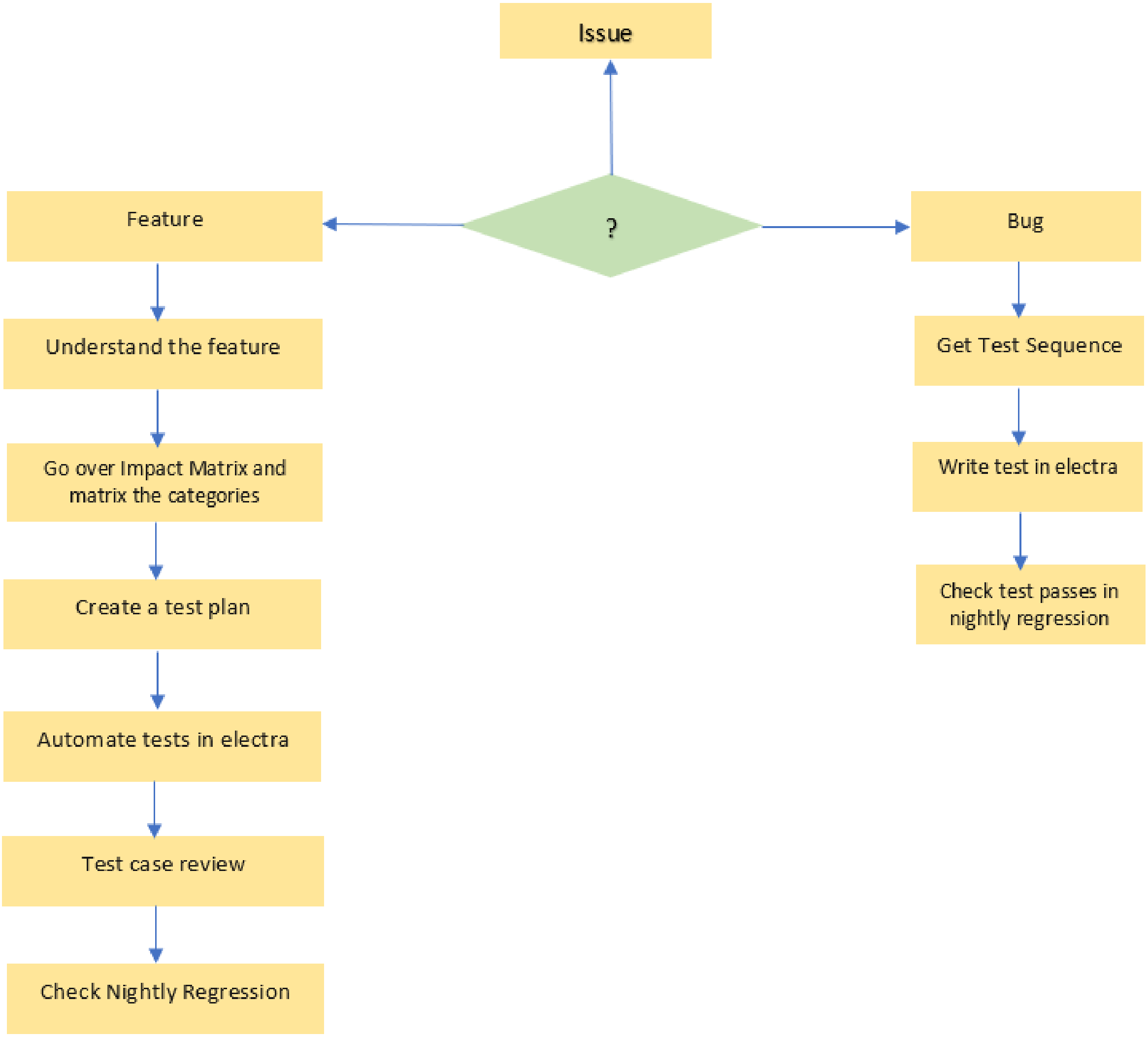
START



REQUEST LIFECYCLE



TESTING FLOW



REFERENCES

- https://intelpedia.intel.com/
- https://www.wikipedia.org/
- https://videoportal.intel.com/

CONCLUSION

START (Storage Tracking, Allocation, and Replication Tool) is a global storage management system which allows it users to create areas specific to projects and users on various fileservers across all Intel sites. And helps easy access, replication and migration of their areas.