



QA/Test Working Group Introduction

Andrew Foster QA/Test WG Chair

Agenda

- QA/Test WG Introduction
 - Scope
 - Testing strategy
 - QA/Test commitments (Edinburgh)
 - QA/Test progress (Edinburgh)
 - Testing resources (wiki, Github, issue tracking, documentation etc.)
- WG organization
 - Current team setup
 - New contributions and organization going forward
 - QA/Test meeting schedule
- Other business

QA/Test WG Scope

1. Testing strategy (tools, frameworks, processes)
2. Test case implementation and documentation (functional, system, performance)
3. Test automation as part of CI/CD (QA/Test works closely with DevOps)
4. Test execution (helping to ensure that tests are running successfully as part of CI/CD)
5. Test maintenance (updating/adding new tests if APIs change, fixing broken tests)
6. EdgeX user documentation – QA/Test originally established EdgeX documentation repo and generation process

Testing Types and Community Involvement

•Unit Test

- Tests developed by individual developers to validate individual component/service capabilities
- Written in same language as component/service e.g. Go or C and using appropriate Unit test framework e.g. CPP Unit for Go Test

•Functional (Blackbox)Test

- Existing tests developed by test team, but has been that each WG can develop functional
- Only run with minimum service dependencies, for example, running Virtual Device Service needs mongo, meta data, and core data.
- Each time the functional test is executed, only the tests belonging to the modified service should be performed.

•System Integration Test

- All the services should be up and run.
- Each time the integration test is executed, all the tests should be performed.
- Test data is generated from virtual device or device simulator
- The backend services contain AWS/Google Cloud/Azure and etc.

•Performance Test

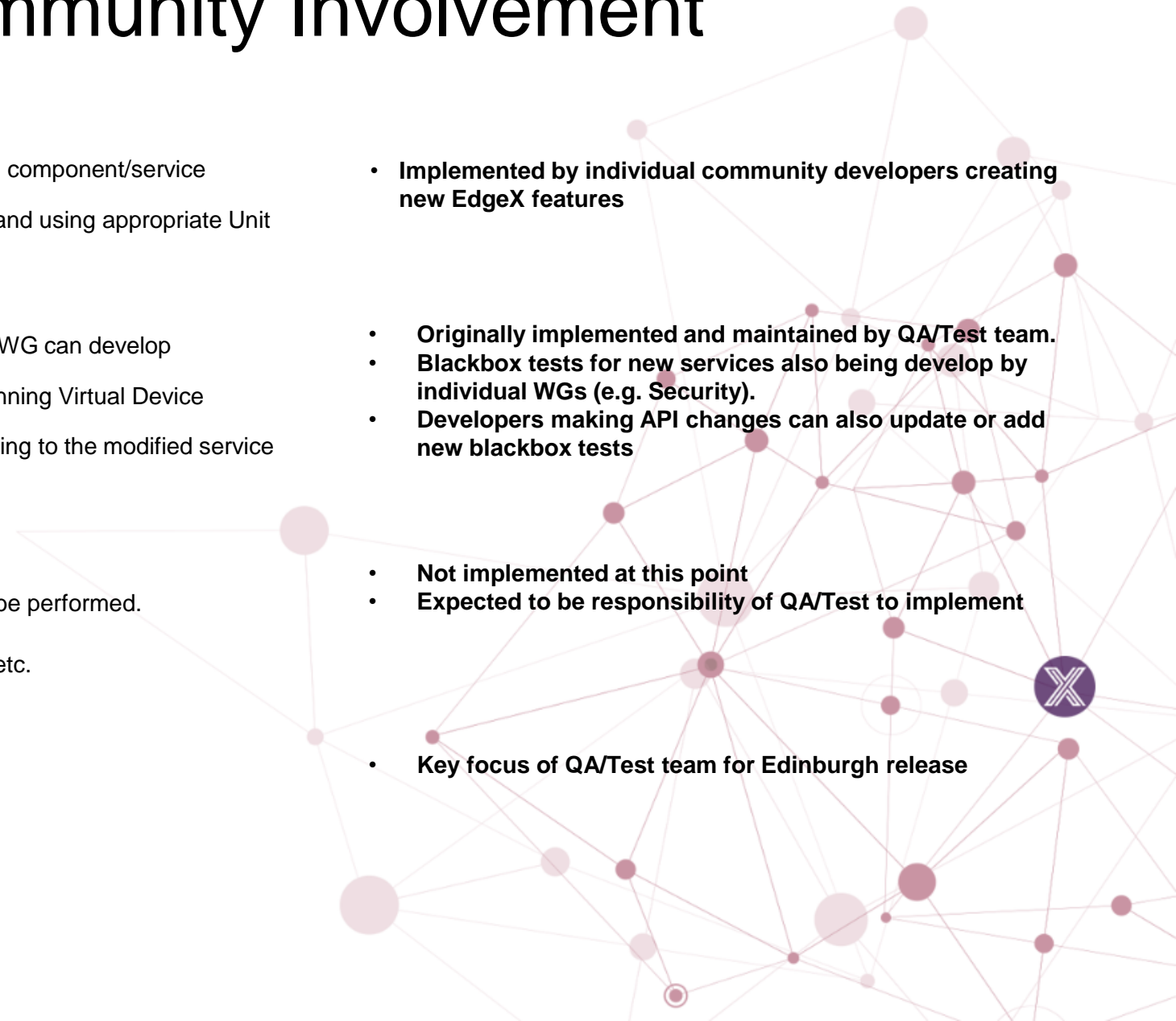
- Monitor CPU/Memory Usage
- Monitor device Request/Response time
- Services startup times
- Latency of data from read to response/export

- **Implemented by individual community developers creating new EdgeX features**

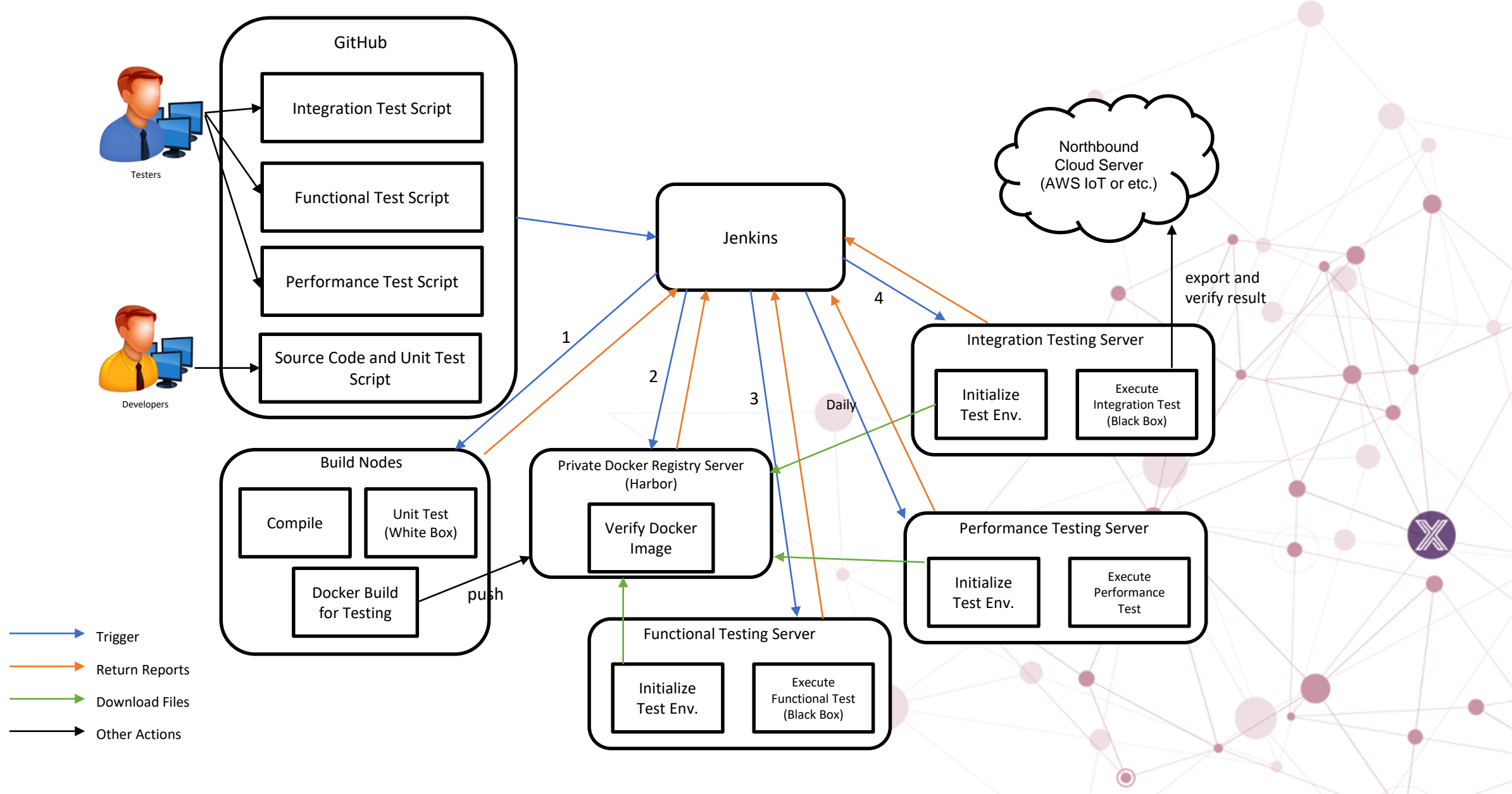
- **Originally implemented and maintained by QA/Test team.**
- **Blackbox tests for new services also being develop by individual WGs (e.g. Security).**
- **Developers making API changes can also update or add new blackbox tests**

- **Not implemented at this point**
- **Expected to be responsibility of QA/Test to implement**

- **Key focus of QA/Test team for Edinburgh release**



Objective CI/CD Driven Automated Testing



Current Testing Tools

- Unit tests – CPP unit, GoTest
- Functional Test Tools - Postman(GUI) and Newman (command line) for REST API testing
- Dashboarding (for blackbox tests) - Allure
- System Integration Test Tool - currently under review
- Performance Test tools
 - Load Testing – Jmeter
 - Monitoring tool (CPU, Memory) – Telegraf + InfluxDB + Grafana

Functional Testing Status

- Objective – automated Blackbox testing of all EdgeX microservices
- Status

EdgeX Microservice	Status	Notes
Core Data	Complete	Already integrated into EdgeX CI process
Core Metadata	Complete	Already integrated into EdgeX CI process
Core Command	Complete	Already integrated into EdgeX CI process
Logging	Complete	Already integrated into EdgeX CI process
Notifications	Complete	Already integrated into EdgeX CI process
Rules Engine	Complete	Already integrated into EdgeX CI process
Scheduling	In progress	Being developed by Core WG (Eric Cotter, Dell) for Edinburgh release
Export client	Complete	Already integrated into EdgeX CI process
Security Services	In progress	Being developed by Security WG (Tingyu Zeng, RSA) for Edinburgh release
Device Services	No Started	Test architecture will defined during Edinburgh iteration

Edinburgh QA/Test Commitments

1. With automated blackbox tests in place, the Edinburgh release will include **better visualization/dashboard of test results** (including look up and display of historical test results). <https://github.com/edgexfoundry/blackbox-testing/issues/105>
2. The Edinburgh release will include the **capture of resource metrics to monitor performance**. Metrics monitored will include memory and CPU consumption. <https://github.com/edgexfoundry/edgex-go/issues/115>
3. As a stretch goal, the Edinburgh release will include **performance/load testing** using a tool like Bender, Jmeter, Load Impact or other tool selected by the work group.
4. **Security tests will be automated** as mentioned above in the Security Group tasks. <https://github.com/edgexfoundry/security-api-gateway/issues/44> & <https://github.com/edgexfoundry/security-secret-store/issues/51>
5. **As a stretch goal, the project will look to produce Swagger documentation** to better support testing and API documentation in future releases. RAML documentation will remain the official API standard for Edinburgh, but additional Swagger documentation will be provided and weighed for use in future releases.
6. For the Edinburgh release, **code contributors will now be required to supply additional or updated blackbox tests** to cover changes made to the code base with any PR. Working group leads and project committers are head accountable for this change in procedures.

Edinburgh QA/Test Progress

Requirement	Status	Notes
Visualization/dashboard of test results	Complete	Allure dashboarding for blackbox test deployed on EdgeX Jenkins Server
Capture of resource metrics to monitor performance	In Progress	Created Telegraf+InfluxDB+Grafana (TIG) stack to provides metrics (CPU, Memory) on EdgeX services. Initial work focused on Core Data and Metadata. For Edinburgh would also like to collect metrics for Core Command and Logging.
Performance/load testing	In Progress	Using JMeter to drive EdgeX APIs. TIG stack used to record API response times and metrics. . Initial work focused on Core Data and Metadata. For Edinburgh would also like to collect metrics for Core Command and Logging.
Automated blackbox testing for Security Services	In Progress	Being developed by the Security WG
Automated blackbox testing for new Go Scheduling Service	In Progress	Being developed by the Core WG
Swagger Documentation	Not Started	Has not been prioritized for Edinburgh
Services startup times	Not Started	Not committed for Edinburgh bur considered a stretch goal not on original list of objectives

Team Setup

- QA/Test team currently consists of WG Chair (Andy) + 4 dedicated (part time) IO Tech resources (Cloud, Cherry, Bruce, Felix)
- Additional contributions for tasks such as Security blackbox testing from Security WG
- Supported by EdgeX DevOps for QA/Test infrastructure deployment
- Contributions from the EdgeX community on testing strategy, tool selection etc.
- New contributions welcome.. who, what, when ?

QA/Test Resources

- QA/Test Wiki - <https://wiki.edgexfoundry.org/pages/viewpage.action?pageId=329484>
- Blackbox testing repository - <https://github.com/edgexfoundry/blackbox-testing>
- How to run blackbox testing documentation - <https://github.com/edgexfoundry/blackbox-testing/blob/master/docs/How-to-run-blackbox-testing.rst>
- Blackbox testing issues - <https://github.com/edgexfoundry/blackbox-testing/issues>
- Performance testing repository - <https://github.com/edgexfoundry-holding/performance-test>
- Performance testing documentation (test creation and environment setup) - <https://github.com/edgexfoundry-holding/performance-test/blob/master/docs/Performance-Test-Environment-and-How-to-Develop-Test-Script.rst>

Other Business

- Codecov.io eval scheduled for next week
- LF can't deploy a TIG stack – they can provide an Elk stack (<https://www.elastic.co/>) which can be integrated with Telegraf, need to understand if this can be used as replacement for InfluxDB+Grafana
- QA/Test meeting schedule – every week, bi-weekly ?



EDGE X FOUNDRY™

Thank You