



EDGE X FOUNDRY™

DevOps Working Group

Thursday July 11, 2019

Agenda

Time	Topic	Owner
10 min	Edinburgh : Dot Release Discussion / Planning	James Gregg / Eric Ball / Jim White
10 min	Performance Testing – LF Configuration EB IAC	Robin
10 Min	Fuji Update	James Gregg
10 Min	GitHub Projects for Issue Tracking	All
	Opens	All

Attendees

Initials	Name	Organization	Video	Audio
JG	James Gregg	(Intel) (Me)	On	On
CT	Cloud Tsai		On	Off
TC	Trevor Conn		Off	Off
AF	Andrew Foster		Off	Off
ER	Emilio Reyes		Off	Off
EB	Eric Ball	Linux Foundation	Off	Off
E	Ernesto	Intel	Off	Off
I	ijohnson		Off	Off
	Jacob Blain Christen	(Intel)	Off	Off
JW	Jim White	(Dell)	Off	Off
LG	Lenny Goodell	(Intel)	Off	Off
R	Ricardo		Off	Off
R	robin		Off	Off
V	Vishwas		Off	Off



EdgeX DevOps WG Update (Edinburgh Dot Release)

- Edinburgh Dot Release (1.0.1)
- An race condition issue was discovered in go-mod-messaging
 - It occurred when core-data sent data to export services (or appl services)
 - It was intermittent, but was verified as an issue by several people
 - It appears most often on Mac OS machines
 - If this error occurs, nothing upstream will get data from EdgeX
- There may be another issue that needs addressed
 - Issue on low power devices??

- List of Artifacts impacted
 - go-mod-messaging (this is the)module that doesn't get released.

All other services need to be released (1.0.1)

- 1.) We need to merge a fix into the master and Edinburgh branches of go-mod-messaging
- 2.) The Edinburgh branch of both edgex-go and app-functions-sdk will need to consume the Edinburgh branch of go-mod-messaging
- 3.) We will need to republish the artifacts for all services in edgex-go for the Edinburgh release as well as app-functions-sdk
- 4.) We will need to adjust the Edinburgh docker-compose files to utilize the version numbers applied to the new images by the above actions.

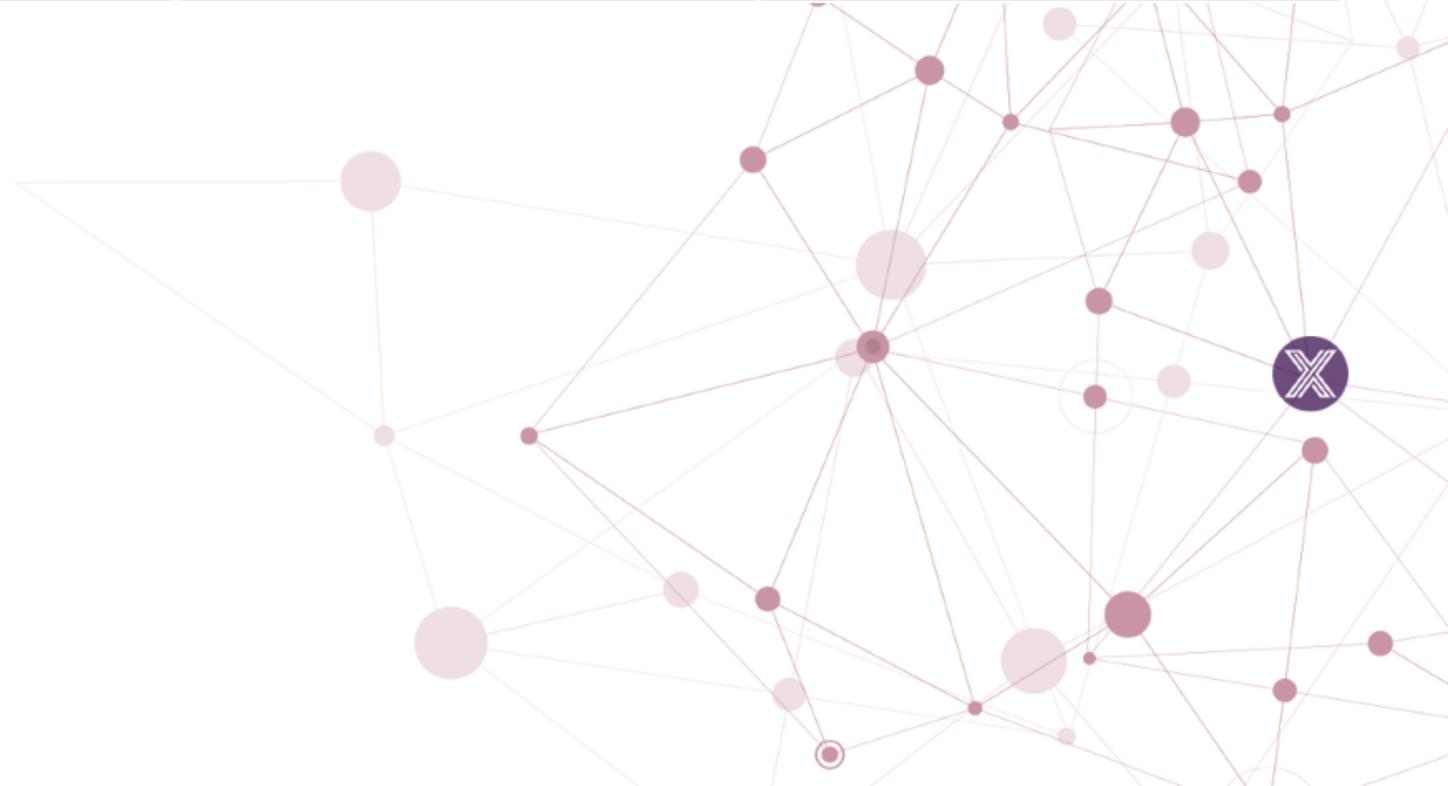
Decision: Not ready to set a date yet... May be other issues to pull into the dot release as well.
Jim White will send email out next Tues with discussion and plan in TSC call on Wed.

DevOps WG Update (recap from TSC call)

- Container Scanning (Clair Server landing request)
Risk for Fuji if the landing doesn't happen
 - Mitigations in place as a interim solution
- Codecov.io integration completed on all repos with tests
- Static Code Analysis Tools - Enabled SonarCloud for EdgeX Foundry - will need to update all JJB templates to enable use of the tool if this is the direction we plan to go.
 - Discussion continued in next slide
- Performance Testing – Robin / Cloud
 - Received configuration of what LF implemented on AWS for the TIG stack on EB
- Edinburgh Docs
 - edgex-go docs – Documentation migrated out of edgex-go
 - Major refactor by Ernesto completed – THANK YOU!!!
 - Status Check - Michael Hall
- Docker Hub Nightly Push (dev)
 - Old ticket since 08/2018
 - Why not build the images if need latest and greatest
 - Not seeing the demand for this and we will defer this for now
 - **Decision**: Close the Ticket and Mark as “Will Not Work”
- GitHub Projects
 - Release Czar Proposal to use GitHub Projects for all Issues as an aide for the release and overview of what's needed for Fuji release.
- Empty GitHub repos
 - Audit completed identified many empty repos in both main org and holding
 - Send audit results to tsc / core / device services mail lists
 - User account and team membership cleanup

Work Review

Helpdesk Ticket #	Description	Details	Status
75648	Dedicated Clair server for EdgeX	Pending decision on strategy for K8s + cost / availability of resources with LF	WIP



EdgeX Foundry Project

Repositories 89 Packages People 60 Teams 89 Projects 14

DevOps WG Updated 6 minutes ago

Filter cards Add cards Fullscreen

Icebox

- regex for isReleaseStream is overly broad
edgex-global-pipelines#26 opened by dweomer
- Snap builds should use unbuffer for better output logging
ci-management#369 opened by anonymouse64
- edgeXSemver leaves command to caller but doesn't allow specifying the version/image to use
edgex-global-pipelines#15 opened by dweomer
bug

New Issues

- Create the automation to create the SDL Release Artifact requested of all Edinburgh Docker images.
ci-management#438 opened by jamesgregg
enhancement fuji
- Refactor docker push scripts to be more generic
ci-management#435 opened by Iranjbar
enhancement
- repository: add description and topics
git-semver#2 opened by dweomer
enhancement

Release Backlog

- jenkins_pipeline_presentation (FUJI)
delete repo once updated documentation completed
Note: Lisa is working on updated Jenkins Pipeline documentation and this repo will potentially be Obsolete once we review / publish that new documentation to the wiki.
Added by jamesgregg
- <https://github.com/edgexfoundry-holding/go-mod-core-security>
ON-HOLD (FUJI)
Added by jamesgregg
- Placeholder for snap global function in edgex-global-pipelines (FUJI)
US4581
US4580
Added by jamesgregg

In Progress

- Clair Server landing - LF Infrastructure or AWS decision and architecture / SS with TSC approval (@jamesgregg)
ci-management#439 opened by jamesgregg
enhancement fuji

QA/Code Review

- WIP: Add sigul signing to golang binaries
ci-management#318 opened by JPWKU
- Changes requested

SonarCloud Discussion (SAST Explore continued)

The screenshot shows the SonarCloud interface for the EdgeX Foundry Project. The left sidebar contains filters for Quality Gate (Passed: 0, Warning: 0, Failed: 0), Reliability (A: 0, B: 0, C: 0, D: 0, E: 0), Security (A: 0, B: 0, C: 0, D: 0, E: 0), and Maintainability (A: 0). The main area shows the project 'edgex-go' with a 'Configure analysis' button. The footer contains copyright information and links to various pages.

- SonarCloud (Code Quality + SAST)

- Very Limited Security Checks for Go
- Other Languages include (mostly web oriented)
 - OWASP Top 10
 - Sans Top 25
 - CWE

- Rules

- <https://sonarcloud.io/organizations/edgexfoundry/rules>

- **Recommendation** is to bring this forward to the Security WG

- Action Item: James will bring to Security WG for review and continuation of the explore

Meeting Minutes

- Jim will send out message next Tues - make the call on TSC call on timing for the dot release
 - 1.) We need to merge a fix into the master and Edinburgh branches of go-mod-messaging
 - 2.) The Edinburgh branch of both edgex-go and app-functions-sdk will need to consume the Edinburgh branch of go-mod-messaging
 - 3.) We will need to republish the artifacts for all services in edgex-go for the Edinburgh release as well as app-functions-sdk
 - 4.) We will need to adjust the Edinburgh docker-compose files to utilize the version numbers applied to the new images by the above actions.
- New Work Items added to Backlog
 - App functions functions - git semver not working
 - Need to create an Issue
 - Device-gps in holding – slated to move out after TSC approval / will need build automation

Edinburgh Retrospective

What went right?

- Communication of when the release was scheduled, was very clear.
- LF and DevOps team cooperating together seemed to work well but with pressure added on top of everyone
- Most artifacts were ready to release in the beginning. It didn't seem like every repo was affected with issues (no extra work)
- Edinburgh Staging view was very helpful
- Great communication and collaboration between Intel DevOps team members and great prioritization
- Prioritization and Organization of the work (assignments and splitting up the work early on in the code release) was helpful

What went wrong?

- EdgeX-UI repos were late code drop
- Lack of a UI WG
- Communication around details was lacking from WG leads
- Need clear definition and understanding from WG leads that have different / independent release cycles
- JSD was introduced in the middle of the release and introduced issues that impacted communications with LF RE
- Availability and Competing priorities of Eric Ball / RE impacted release work
- Branches cut early caused extra work for both developers and DevOps

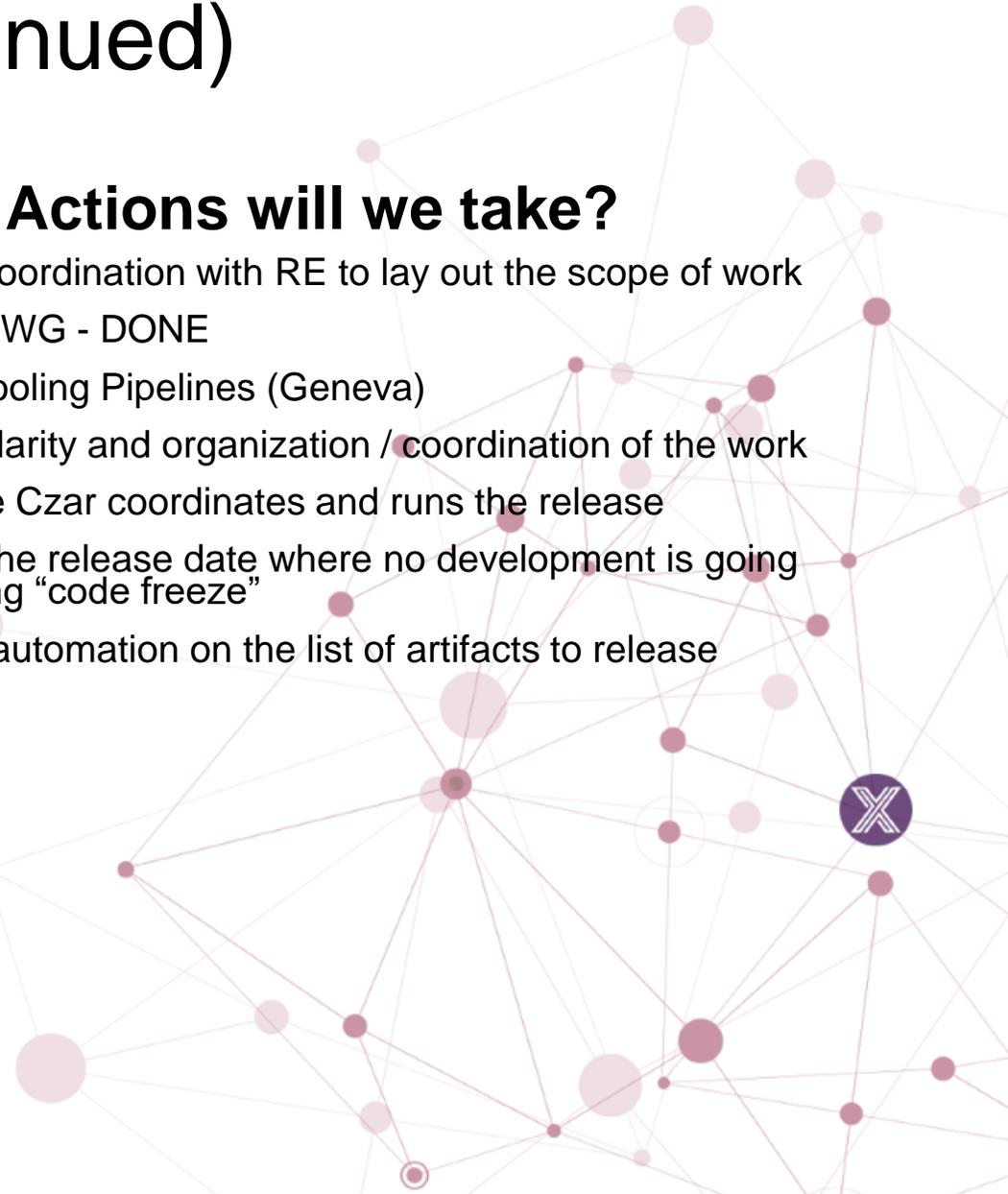
Edinburgh Retrospective (continued)

What Ideas would help next time?

- Jenkins Pipelines branch defined in Jenkinsfile
- Do Not cut the branch early
- Release from master
- Don't allow PRs to master during code freeze (unless bug fix)
- Release Czar manages the release and acts as coordinator
- Need better visibility as to WIP during release. What does DONE look like?
- Set clear expectations for FUJI ahead of time
- Have a solid and well defined scope for executing the release.
- Shorten the release timeframe
- Actually Freeze Code
- Don't pull in late code drops

What Actions will we take?

- Better coordination with RE to lay out the scope of work
- New UI WG - DONE
- Better tooling Pipelines (Geneva)
- Better clarity and organization / coordination of the work
- Release Czar coordinates and runs the release
- Shrink the release date where no development is going on during "code freeze"
- Create automation on the list of artifacts to release





EDGE X FOUNDRY™

Fuji Planning

Scope Discussions

Fuji – DevOps

In

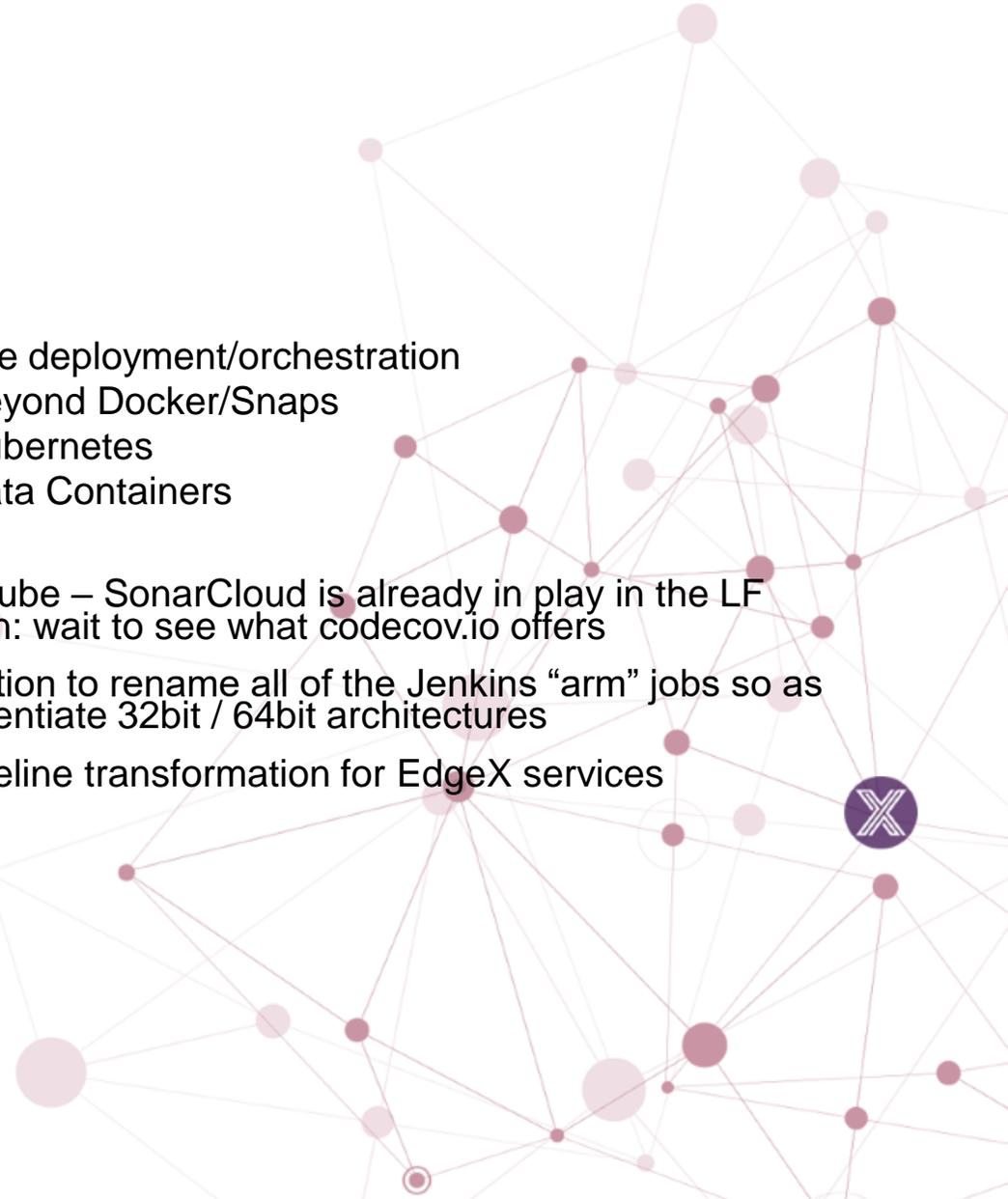
- Static code analysis tool identified and integrated into the EdgeX Jenkins Pipeline for Docker image scanning (Clair Server)

Explore SAST for true static code analysis to include additional tooling such as Fortify / Coverity

- Code and artifact signing with semantic versioning
- Fix Documentation – edgex-go
 - Create a new repo for edgex-docs
- Build Performance Optimizations
 - Pipelines for EdgeX Foundry base build images
 - Basebuild images managed locally within Nexus
 - Leverage PyPi Proxy for local pip dependencies
 - ARM builds – optimization leveraging different high CPU build nodes / OS (ARM Team)

Out

- Alternate deployment/orchestration
 - Beyond Docker/Snaps
 - Kubernetes
 - Kata Containers
 - ...
- SonarQube – SonarCloud is already in play in the LF Decision: wait to see what codecov.io offers
- Suggestion to rename all of the Jenkins “arm” jobs so as to differentiate 32bit / 64bit architectures
- Full Pipeline transformation for EdgeX services



EdgeX DevOps Commitments (Fuji)

Scope of Work	
Add static artifact analysis into the EdgeX Jenkins Pipeline (analysis of Docker /runtime artifacts, not the source code)	
Add code and artifact signing with semantic versioning	
Conduct build performance optimizations by: <ul style="list-style-type: none"> • Adding Pipelines for EdgeX Foundry base build images • Allow base build images to be managed locally within Nexus • Leverage PyPi Proxy for local pip dependencies 	
Explore static code analysis like Checkmarx, Coverity, GuardRails, Synk, SonarQube	

- Clair Server landing at **Risk** for Fuji

- Work Around will be to use Intel hosted Clair server until decision is made by LF to support landing dedicated infrastructure
- Work related to automating the scans as part of the build, will defer to Geneva scope

- gitsemver along with lftools used for artifact signing and semantic versioning

- Jenkins build performance optimizations for base build images completed

- All base build images will now be stored in Nexus (Snapshot):10003

- PyPi enabled as part of Edinburgh scope

- Initial review of GuardRails showed that the product was identifying issues which were not applicable for microservices architecture



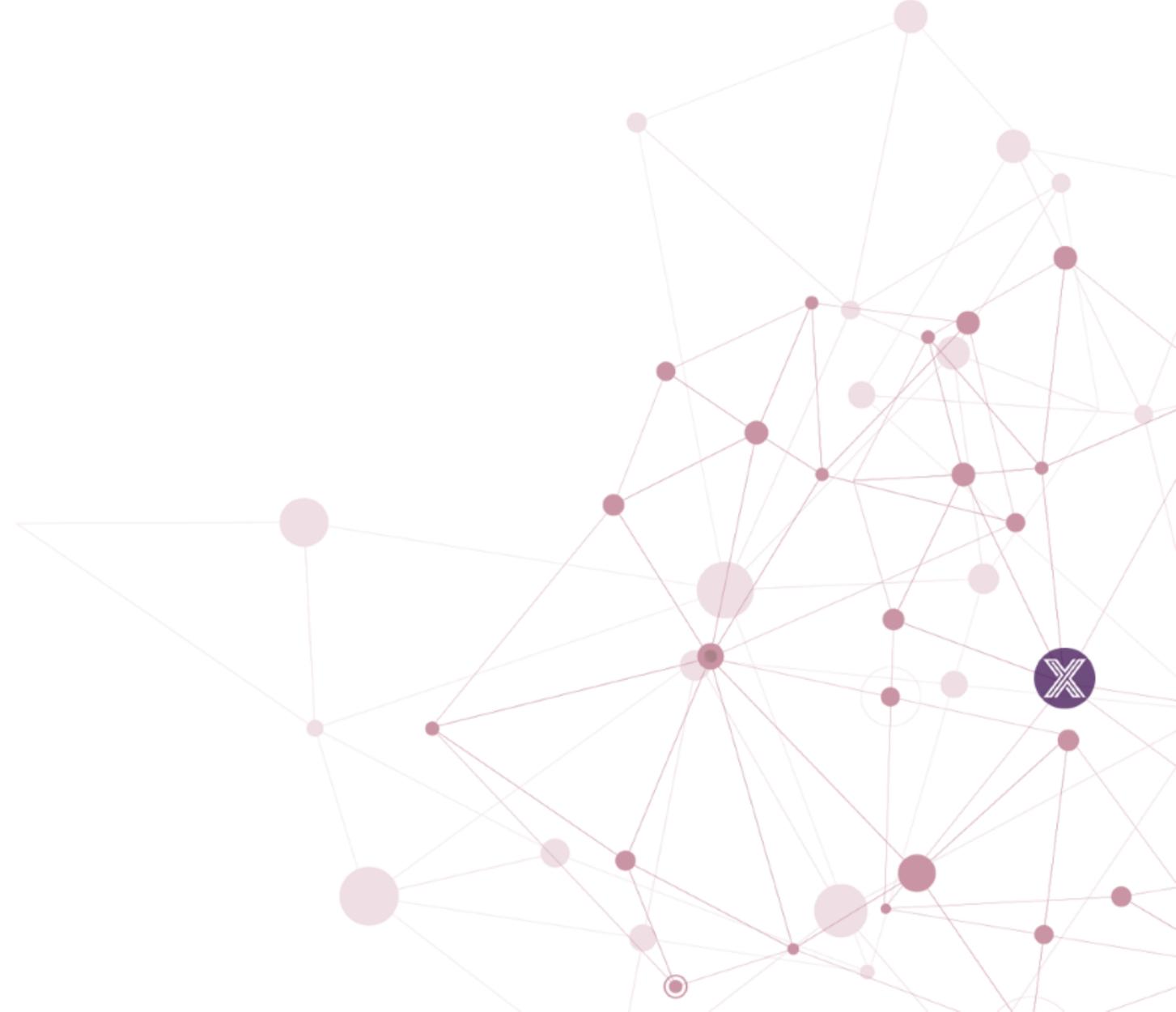
EDGE X FOUNDRY™

Edinburgh Release

Release Planning

Edinburgh Dates

- Freeze Date – May 28
- Release Date – June 20
- Press Release – July 11
- Dot Release – July ?





EDGE X FOUNDRY™

Past / Future Agenda Topics

WW27	No Meeting – US Holiday
WW28	
WW29	
WW30	
WW31	
WW32	
WW33	
WW34	
WW35	
	Athens Project – proxy server for go package dependencies
	Community Involvement