# CxFLOW

Training

Checkmarx

THE WORLD RUNS ON CODE. WE SECURE IT.

# INTRODUCTION

- CxFlow was originally developed by Ken McDonald when he was at Custodela
- CxFlow can be used to integrate CxIAST, CxSAST and CxSCA with external systems
  - CI/CD platforms can use CxFlow to trigger scans and process scan results
  - CxFlow can run as a server that can process Webhook calls from SCM platforms
  - CxFlow has support for many bug tracking systems and feedback channels
- CxFlow is a Spring Boot application
  - Can run anywhere Java can run (also available as a Docker container)
  - Java 8 and Java 11+ supported

# INTRODUCTION

- Developed on GitHub: https://github.com/checkmarx-ltd/cx-flow
  - Apache license (version 2.0)
- The CxFlow roadmap is in Aha!: https://checkmarx1.aha.io/products/SDLC/feature_cards
- Depends on the Checkmarx Spring Boot Java SDK (also developed in GitHub: https://github.com/checkmarx-ltd/checkmarx-spring-boot-java-sdk)

# CXFLOW MODES OF OPERATION

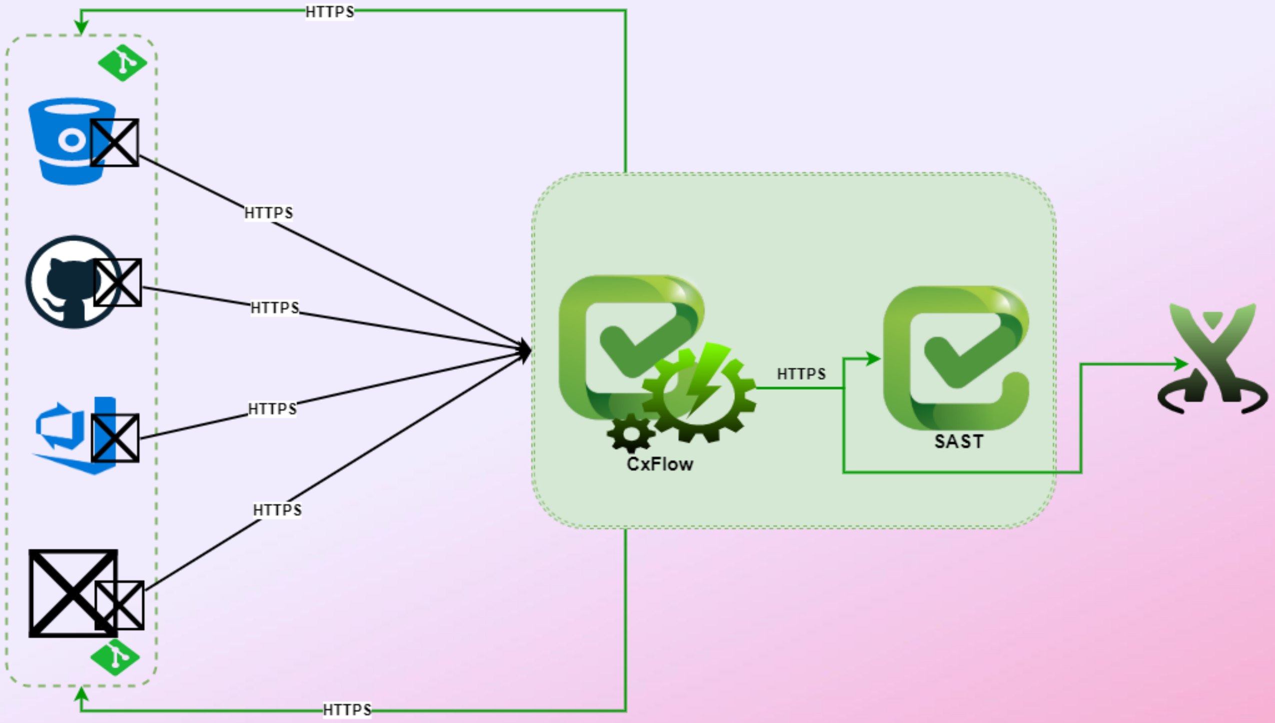
- CxFlow has two modes of operation:
  - Batch mode
  - Server mode

# CXFLOW IN BATCH MODE

- When run in batch mode, CxFlow can:
  - Trigger a new scan (`--scan` command line option)
  - Process results of existing scans
    - Either for a specific project (`--project` command line option) or for all projects (`--batch` command line option)
  - Parse a SAST XML report file (`--parse` command line option)

# CXFLOW IN SERVER MODE

- In server mode, CxFlow can process Webhook requests from:
  - Azure DevOps
  - Bitbucket
  - GitHub
  - GitLab



# CONFIGURATION

- CxFlow can be configured via the following mechanisms:
  - A YAML configuration file (`application.yml`, by default)
  - Command line options
  - Environment variables
- In batch mode, CxFlow supports many additional command line options
- In server mode, and batch mode, when scanning local source, a config-as-code file can be used

# CONFIG-AS-CODE

- A `cx.config` file in a project's root directory can be used to override certain configuration settings on a per-project basis
- See https://github.com/checkmarx-ltd/cx-flow/wiki/Config-As-Code

# BUG TRACKERS AND FEEDBACK CHANNELS

- See https://github.com/checkmarx-ltd/cx-flow/wiki/Bug-Trackers-and-Feedback-Channels
- Two special bug trackers:
  - NONE – CxFlow will not wait for the scan to complete (i.e., can be used to launch asynchronous scans)
  - WAIT – CxFlow will wait for the scan to complete but will not process the results

# PRE-PACKAGED CI/CD SUPPORT

- Checkmarx provides:
  - A CxFlow GitHub Action
    - https://github.com/checkmarx-ts/checkmarx-cxflow-github-action
  - A GitLab CI/CD template
    - See https://checkmarx.com/resource/documents/en/34965-8218-gitlab-integration.html

# TROUBLESHOOTING

- Double-check the command line
  - CxFlow will silently ignore incorrect command line options
  - All options (even single letter options) have a double-hyphen prefix (e.g., “--f”,  not “-f”)
  - Option arguments are separated from their options by an equals sign (e.g., “--f=src”, not “--f src”)
- Make sure correct Checkmarx SAST version specified
  - Defaults to 8.x
- Change logging configuration
- See https://github.com/checkmarx-ltd/cx-flow/wiki/Troubleshooting

# CXFLOW RELEASES

- Compiled jar files for Java 8 and for Java 11+ are available from the CxFlow GitHub releases page
    - https://github.com/checkmarx-ltd/cx-flow/releases
- A Docker image is available from DockerHub
    - https://hub.docker.com/r/checkmarx/cx-flow

# DOCUMENTATION

- There is a CxFlow wiki on GitHub
  - https://github.com/checkmarx-ltd/cx-flow/wiki
- The GitLab CI/CD integration is documented in the product documentation
  - https://checkmarx.com/resource/documents/en/34965-8218-gitlab-integration.html

# THANKYOU

James.Bostock@checkmarx.com | www.checkmarx.com

THE WORLD RUNS ON CODE. WE SECURE IT.