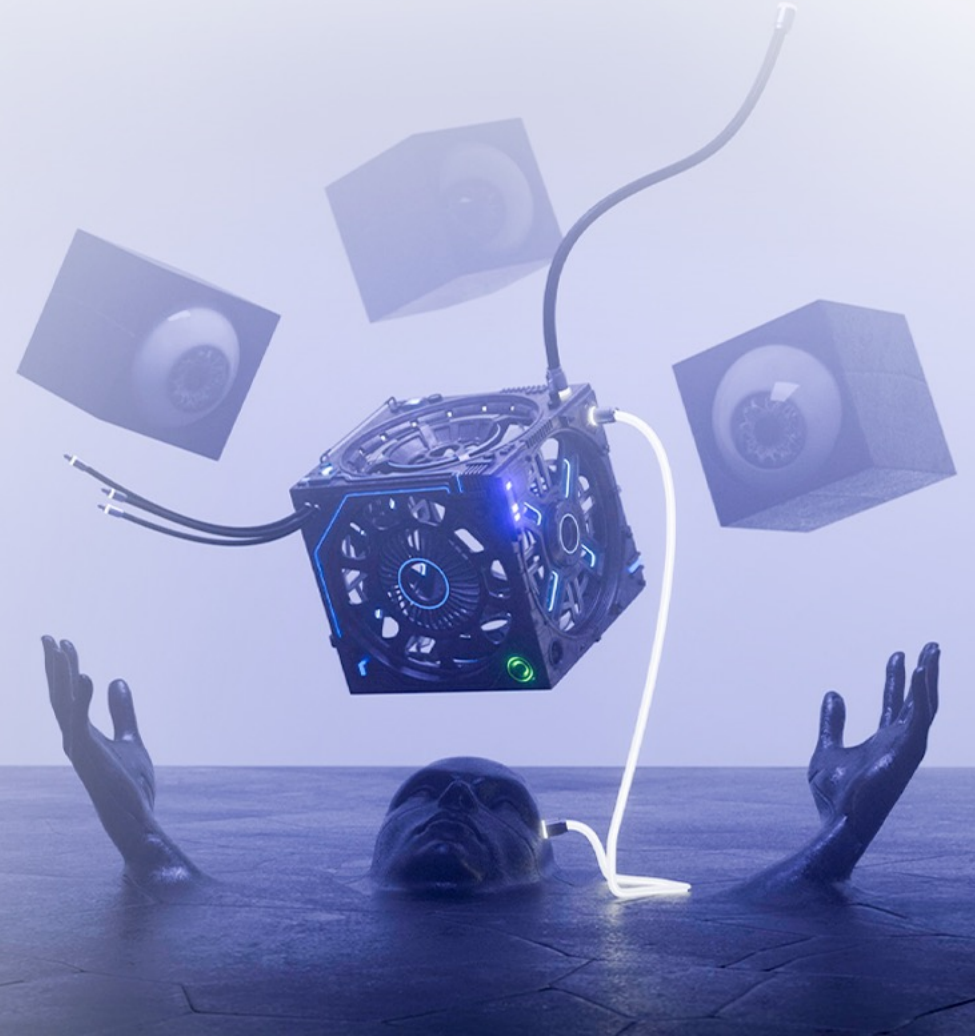
ON FF ONE 2023

STINGRAY

5 Lifehacks for Mobile DevSecOps



Yury Shabalin

CEO, Stingray Technologies

@Mr_R1p

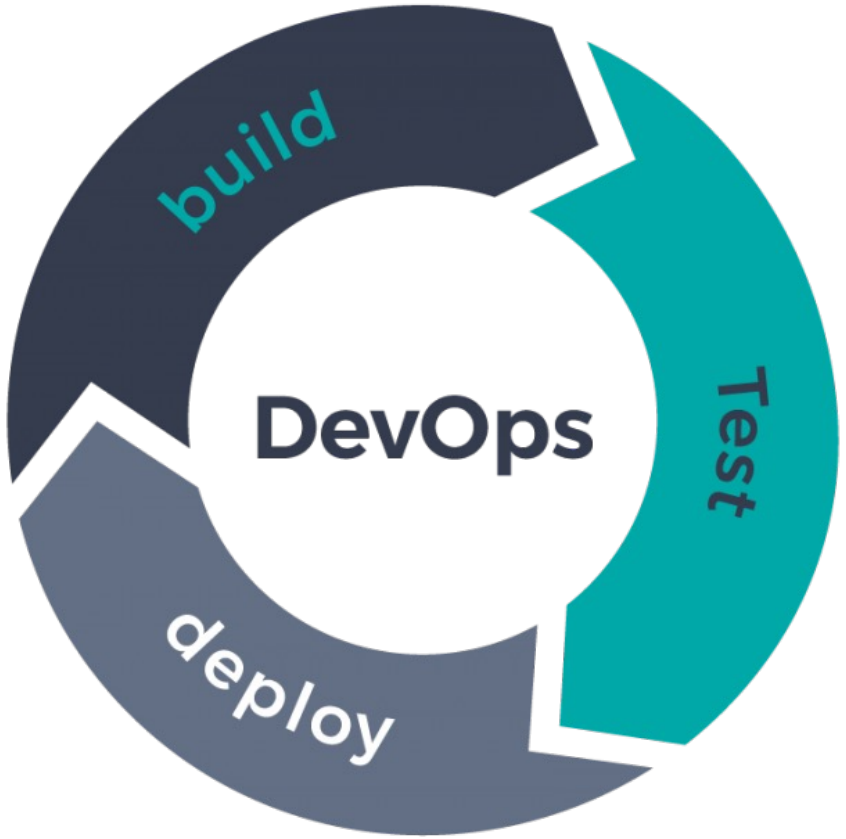
Who Am I

Yury Shabalin

- CEO at ”Stingray Technologies LLC”
- Senior Security Architect at “Swordfish Security LLC”
- Mobile Application Pentester
- Security Researcher
- DevSecOps Evangelist

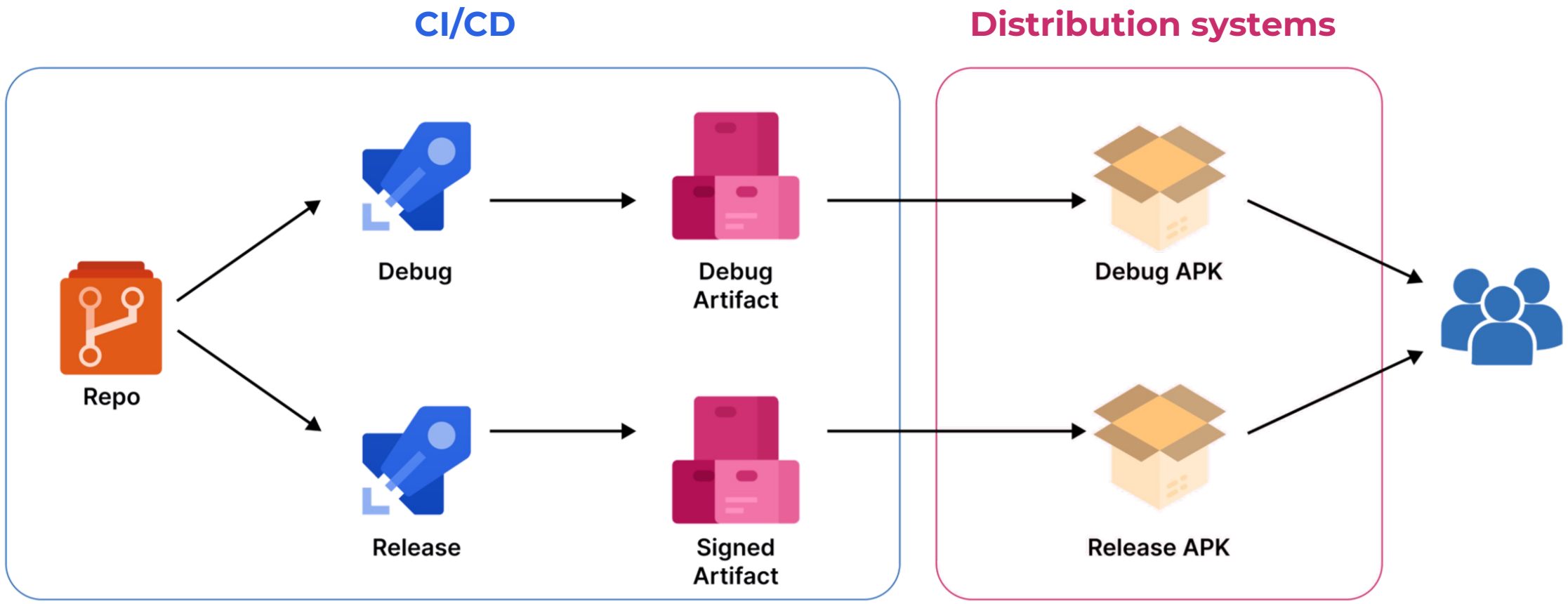
Standard process

Commit, Push, Build, Deploy, Test, Deploy to production (?)



Mobile process

Commit, Push, Build, Distribute, Install on devices, Manually upload to Store



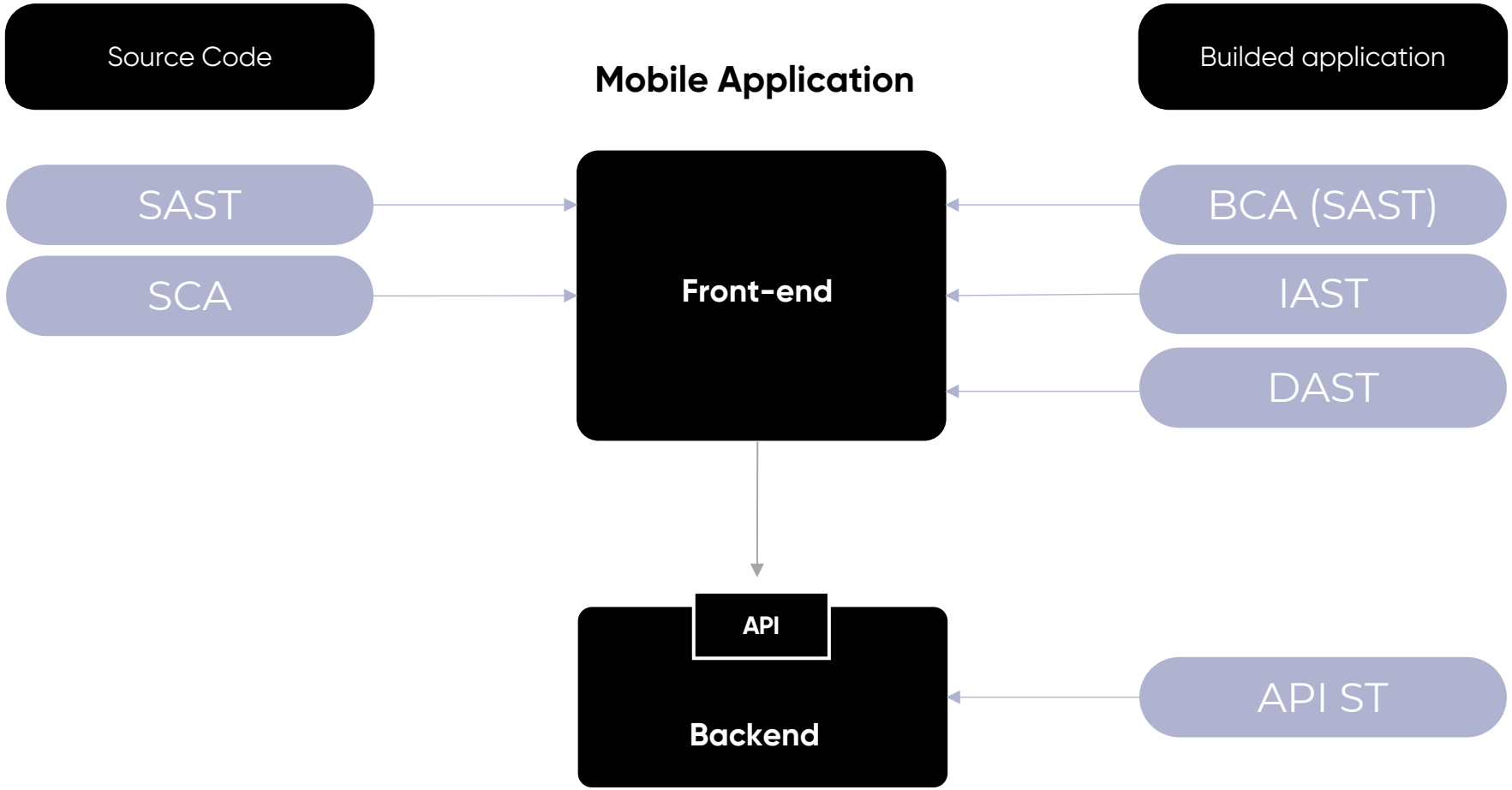
Mobile process

- The development process is often outside the company infrastructure
- The final artifacts are stored in special Distribution systems such as GooglePlay Console / Firebase, AppCenter, Testflight, etc)
- Test servers - simulators, emulators, numerous real devices
- Sometimes different builds for different environments or purposes
- Feature Toggles 🫠
- Work in the most terrible and uncontrolled environment
- Fast releases (sometimes not depending on Backend release cycle)

Special Security Requirements for Mobile

- Increased requirements for code protection (obfuscation, encryption)
- And to keeping secrets too
- Environment Checks
- Application Integrity Check
- Test data (Test credentials, testing URLs, etc)
- Requirements are very different to Web Applications
- CrossPlatform… 

Default Security Practices



SAST (Static Application Security Testing)

- No Special Requirements, but….
- Many tools doesn’t have rules for specific vulnerabilities in mobile apps
- Some tools are not keeping up with dynamically developing technologies in mobile applications

SCA (Software Composition Analysis)

- Can only be used on source code
- Sometimes it is hard to build SBOM file
- No other special requirements

BCA (SAST) (Byte Code Analysis)

- Based on decompiled/disassembled application
- Shows what exactly got into the builded application and inside the bundle
- May be hard due to obfuscation or other protection mechanisms

IAST (Interactive Application Security Testing)

- Applied on running application
- Defines data flows in a real running application
- Identifies the shortcomings of an Open Source component
- Determines the quality of environmental checks and other security measures
- Specifies the transfer of sensitive information to Third Party Services

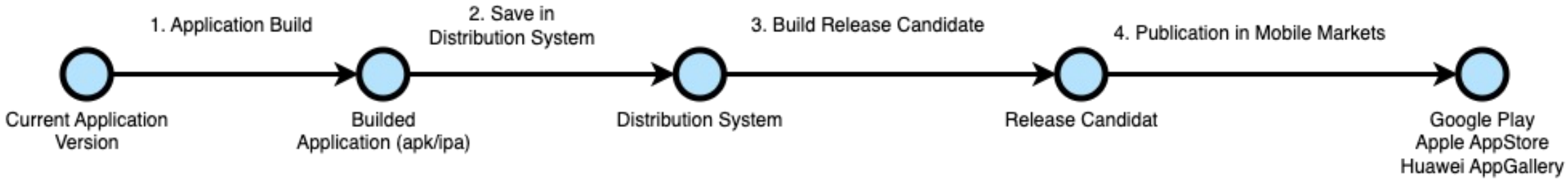
DAST (Dynamic Application Security Testing)

- Defines all available application entry points (Applink, Deeplink, Activities, Content Providers, etc)
- Interacts with the application as a user or a nearby application
- Fuzzing input points
- Complements IAST

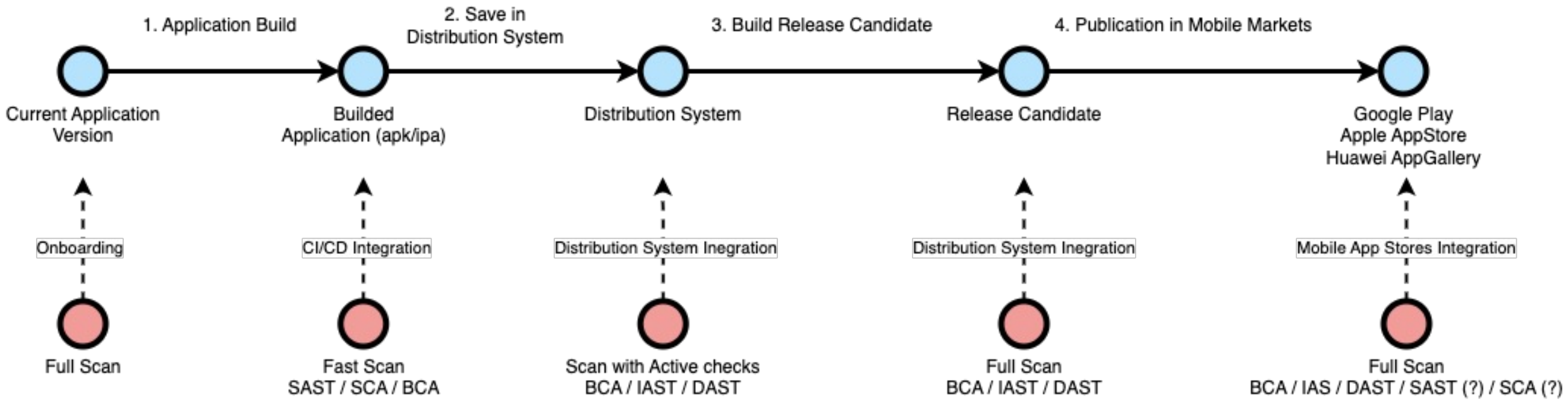
API ST (API Security Testing)

- Getting network traffic between client and server
- Collecting data about the structure of the API
- Backend analysis based on real traffic
- Enriching profile tools to improve API testing (Burp Suite, Acunetix, Netsparker, etc)

Simplified Mobile Development Process

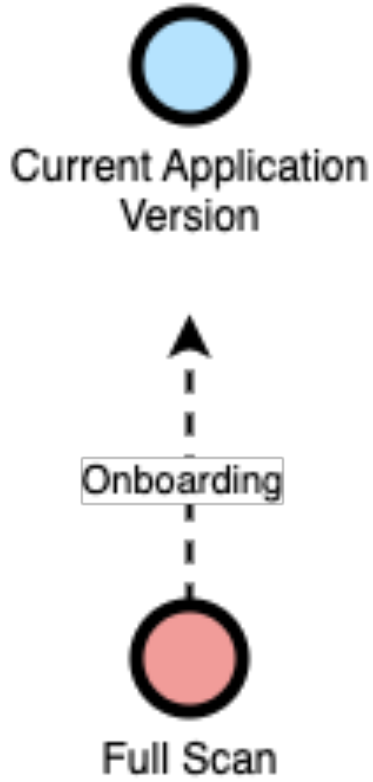


Simplified Secure Mobile Development Process

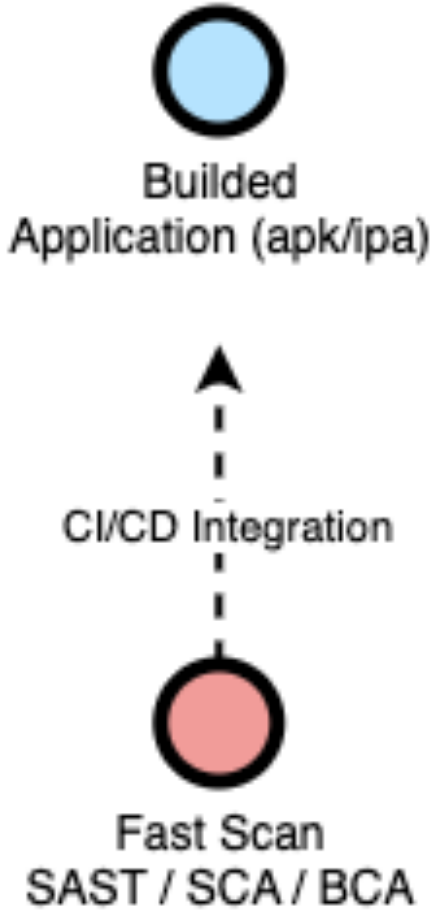


Onboarding

- Full Scan with all tools and practices
- Tools fine tuning
- Understanding the development process
- Manual and auto scans.
  One time-action
- Next steps for integration



CI/CD Integration

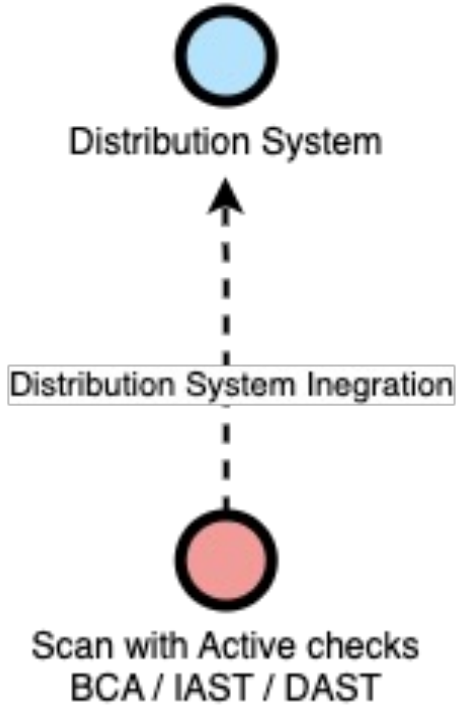


- Integration in CI/CD Process
- Fast scan with SAST and SCA Tools
- Has little effect on build time
- Quick understanding of the current situation and feedback to development
- Auto scans. Repeat every build

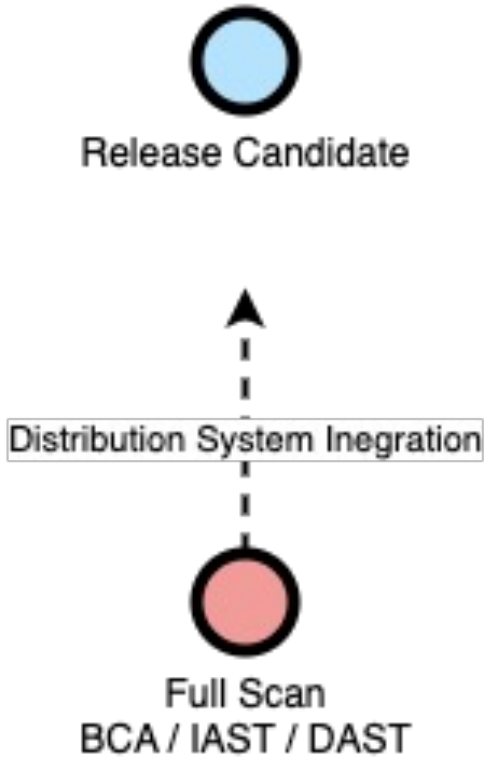
Distribution System

- Integration with distribution systems
- Get the result builded file as testers do
- Does not affect the build process and build time
- Work in parallel with functionality testing
- All Active Checks included

- Auto scans. Repeat every build



Release Candidate

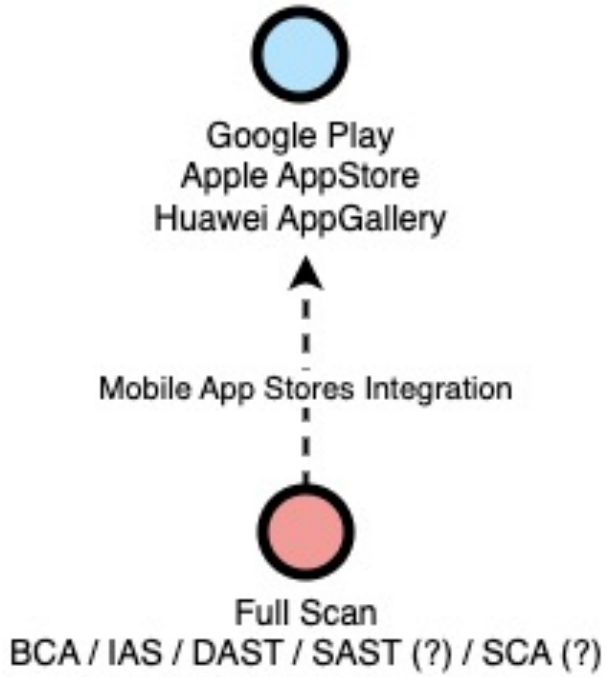


- Testing all new functionality in release
- Testers may go throw their test-plans of this release
- Has little effect on release time
- All new functionality covered via security checks
- Manual and auto tests. One time per release.

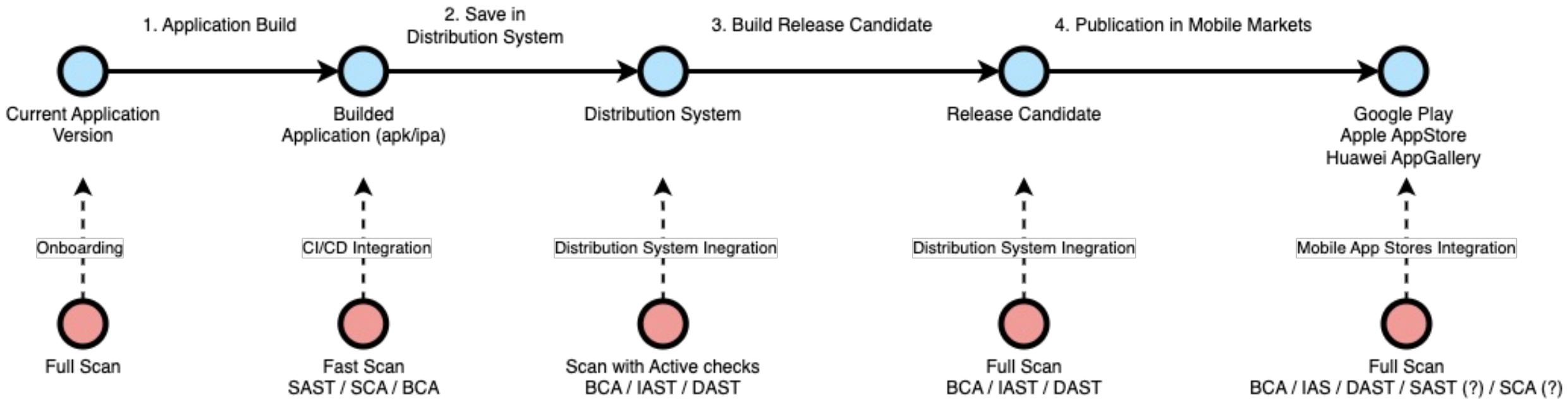
Mobile Application Stores

- Integration with mobile Application Stores
- Check the release build
- Use your own account or without it
- All Active Checks included

• Repeat every release



Simplified Secure Mobile Development Process



More advices not covered in presentation

- Use automation everywhere where it is possible
- Re-use auto-tests from your QA team for interaction with application UI to generate data for DAST/IAST/API ST practices
- Save network communication between application and backend to improve standard DAST tools
- Try to use distribution system as much as possible – it is very convenient.
- Don’t forget to check Release version after publishing to stores

Tool for integration with Distribution Systems and Stores:
https://github.com/Dynamic-Mobile-Security/mdast-cli

Lets discuss about Mobile Security

- Join to our ”Mobile Appsec World” Telegram Channel
  https://t.me/mobile_appsec_world

- Find our stand in Community.Zone on OFFZONE 2023!


ON
FF
ONE
2023