CHICAGO
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE 2016

goto; conference

Building a Distributed Build System at Google Scale

Aysylu Greenberg

Aysylu Greenberg

@aysylu22

Aysylu Greenberg

Google

f(x) = x

@aysylu22

Building Distributed Build System at Google Scale

Building Distributed Build System *at Google Scale*

WTH is "Google scale"?

Google Scale

- Engineers: >30,000 developers in 40+ offices

Google Scale

- Engineers: >30,000 developers in 40+ offices
- Commits: 15K by humans + 30K by robots per day

Google Scale

- Engineers: >30,000 developers in 40+ offices
- Commits: 15K by humans + 30K by robots per day
- Source code: 2 billion LOC

Google Scale

- Engineers: >30,000 developers in 40+ offices
- Commits: 15K by humans + 30K by robots per day
- Source code: 2 billion LOC
- Builds and tests: 5M per day through BuildRabbit

Google Scale

- Engineers: >30,000 developers in 40+ offices
- Commits: 15K by humans + 30K by robots per day
- Source code: 2 billion LOC
- Builds and tests: 5M per day through BuildRabbit
- Petabytes of output artifacts

Google Scale

- Engineers: >30,000 developers in 40+ offices
- Commits: 15K by humans + 30K by robots per day
- Source code: 2 billion LOC
- Builds and tests: 5M per day through BuildRabbit
- Petabytes of output artifacts
- 1 repository

Working in One Repository

Working in One Repository

- Linear revision history

Working in One Repository

- Linear revision history
- Everything is cross-referenced

Working in One Repository

- Linear revision history
- Everything is cross-referenced
- Components for library releases
    - = Git subtree or Git subcomponents to separate release from WIP versions

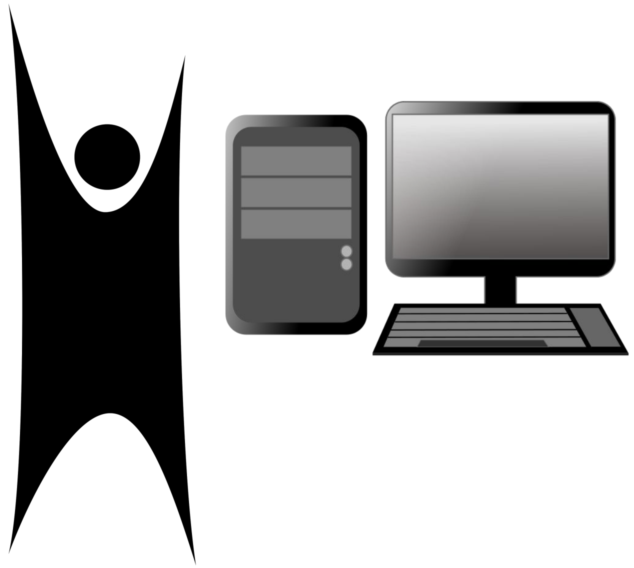
Working in One Repository

- Linear revision history
- Everything is cross-referenced
- Components for library releases
- Repository of artifacts vs build from source:
  - Predictable, repeatable builds from source
  - Optimizations to avoid compiling same artifacts
  - Decouple each team's processes as much as possible

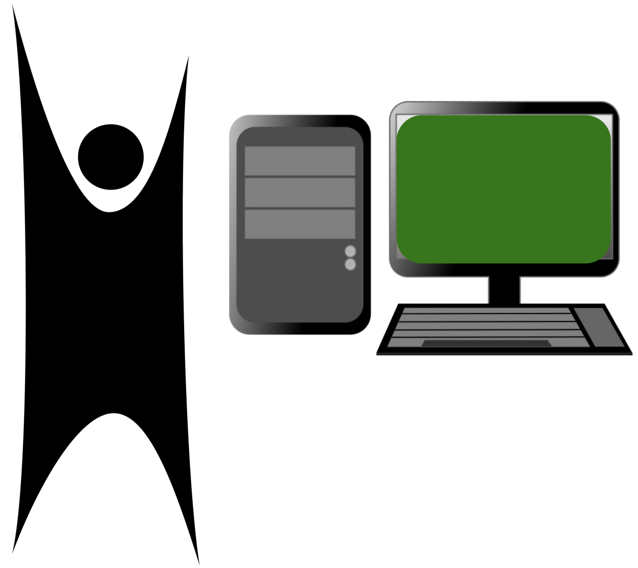
Building Distributed Build System at Google Scale

Building
Distributed Build System
at Google Scale

Towards Distributed Build System

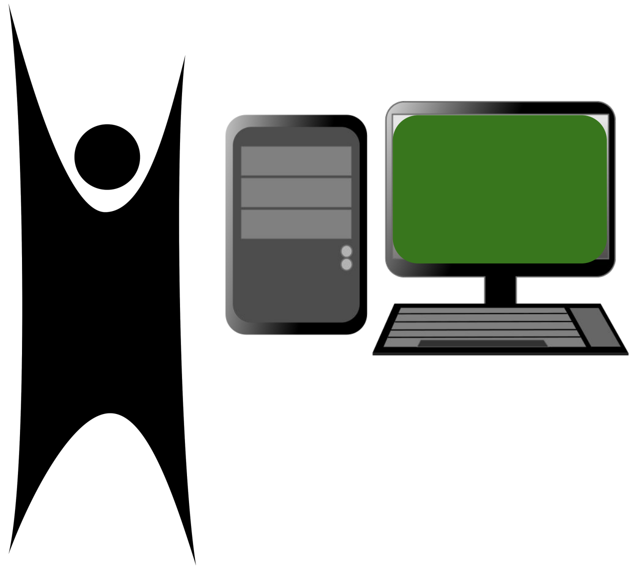


Towards Distributed Build System

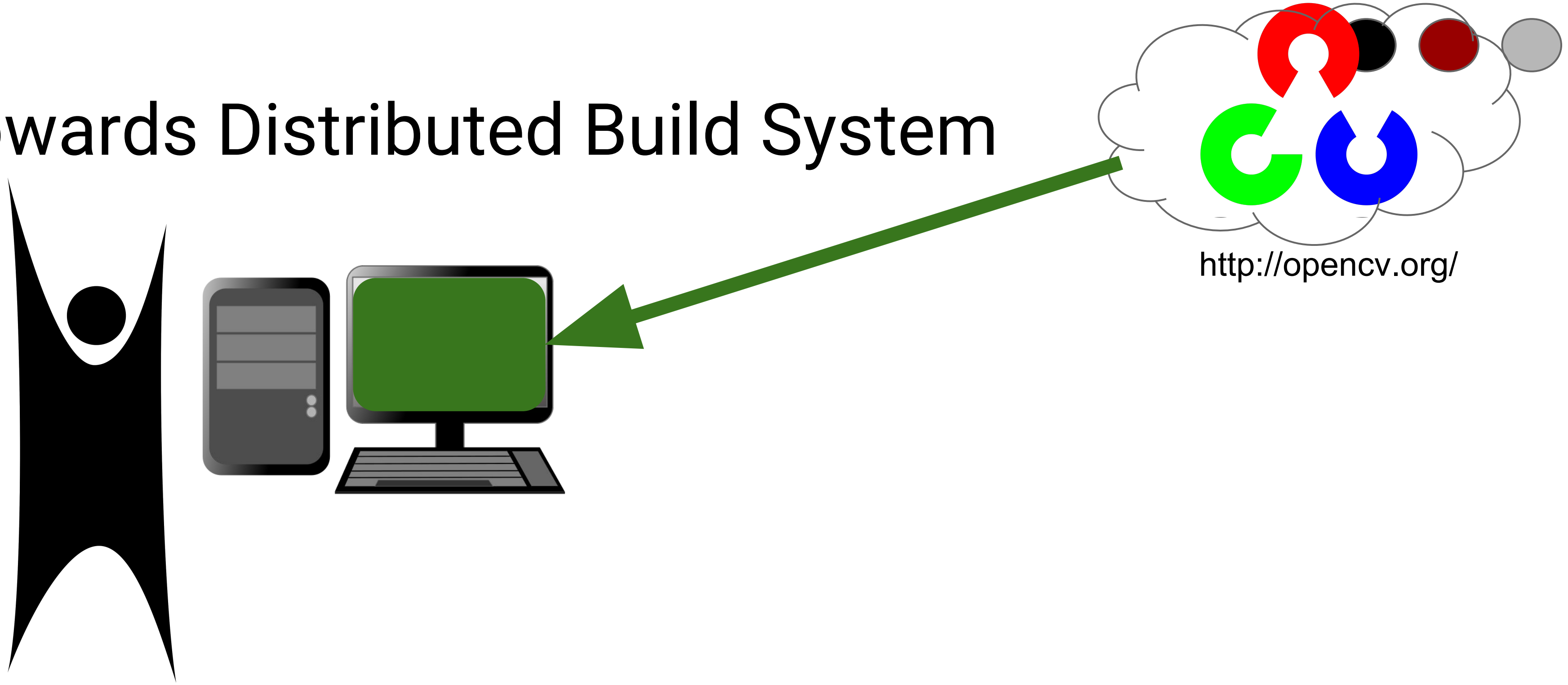


Towards Distributed Build System

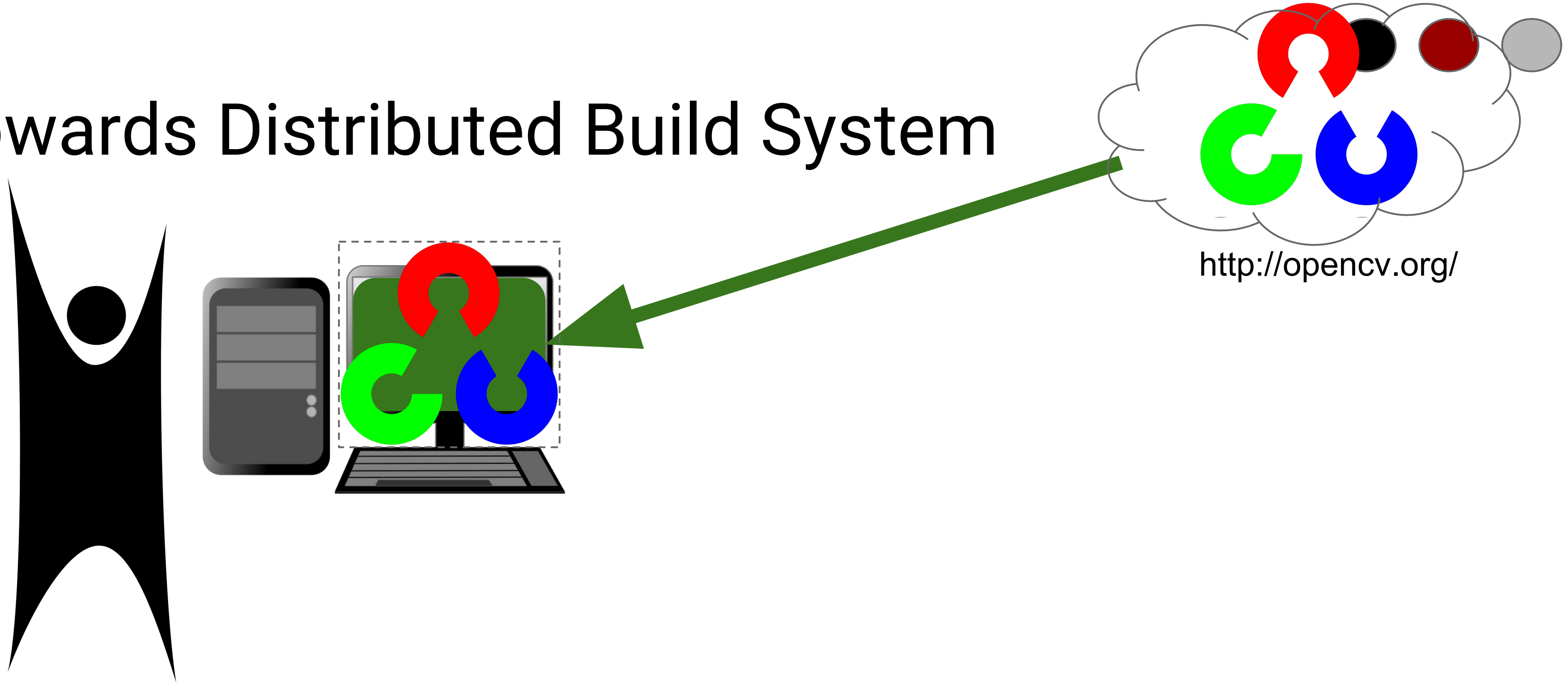
http://opencv.org/



Towards Distributed Build System



Towards Distributed Build System



Build Scenario:

Project with dependencies

Find dependencies

Build project with the dependencies

Download build artifacts

Test Scenario:

Project with dependencies

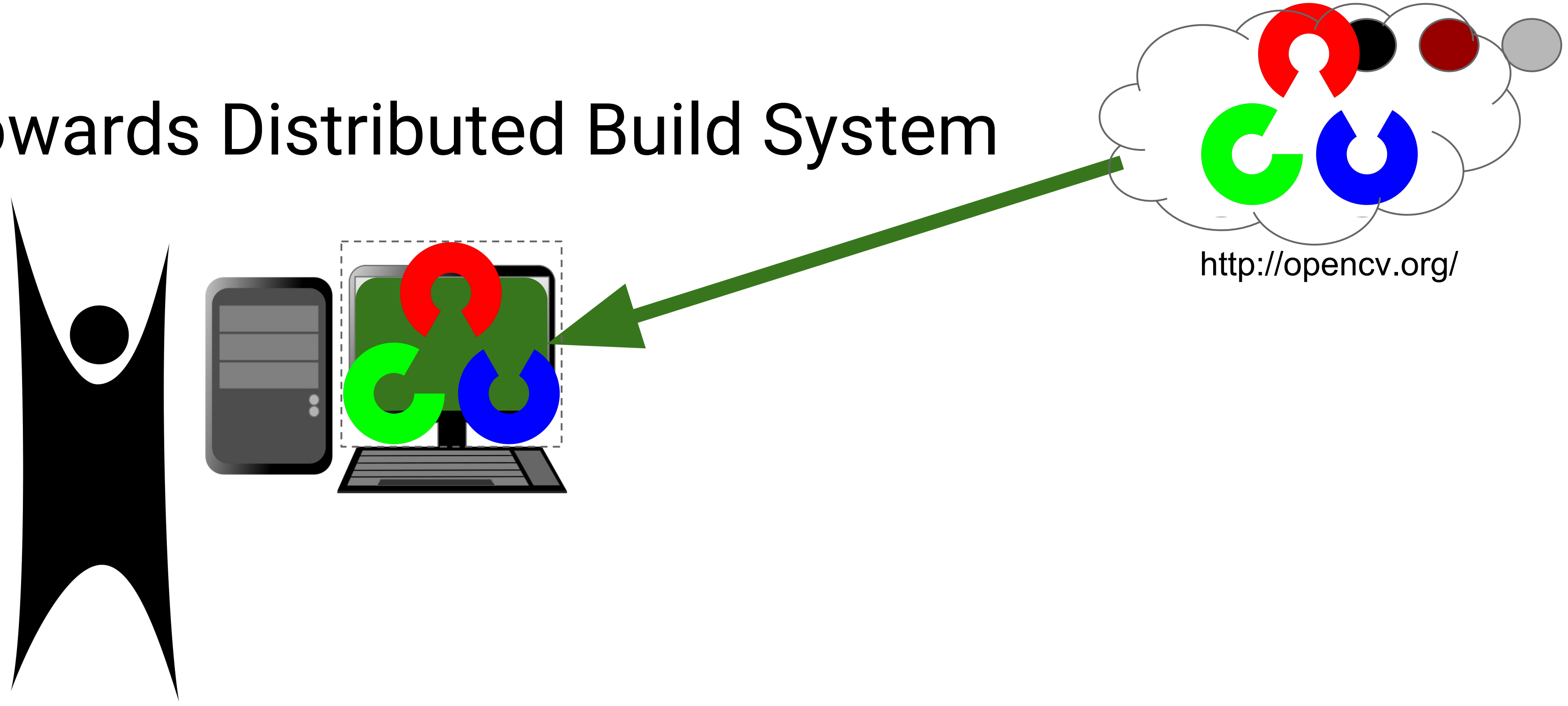
Find dependencies

Build project with the dependencies

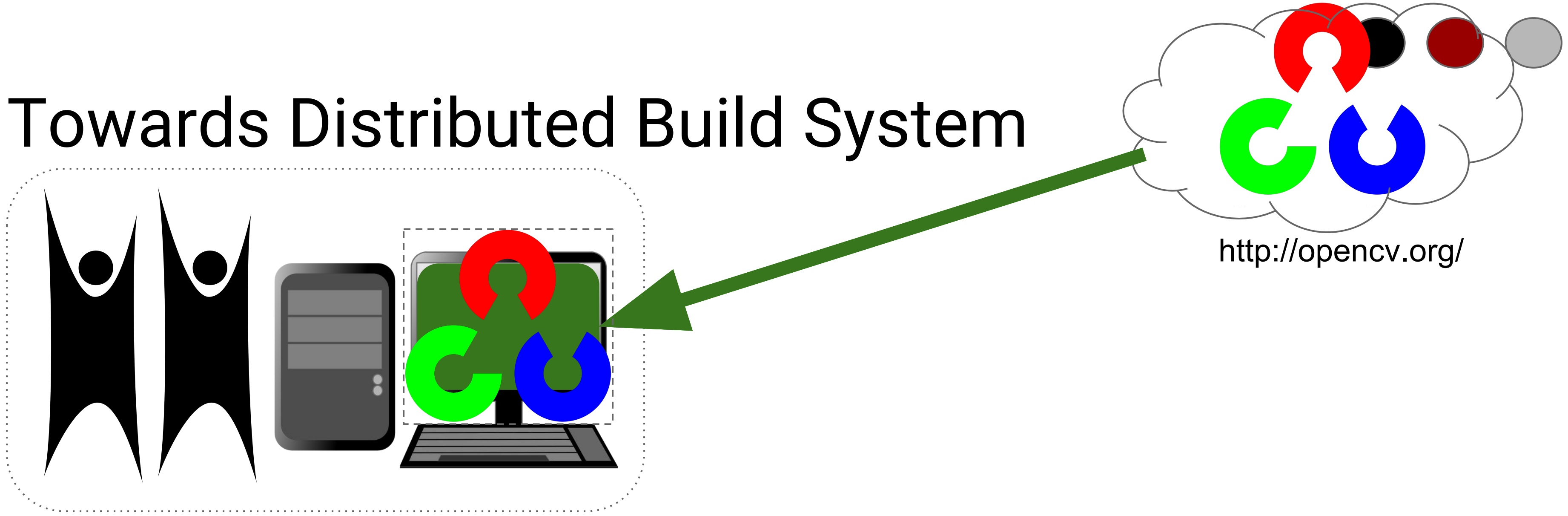
~~Download build artifacts~~ Run the test

Get the results of the test

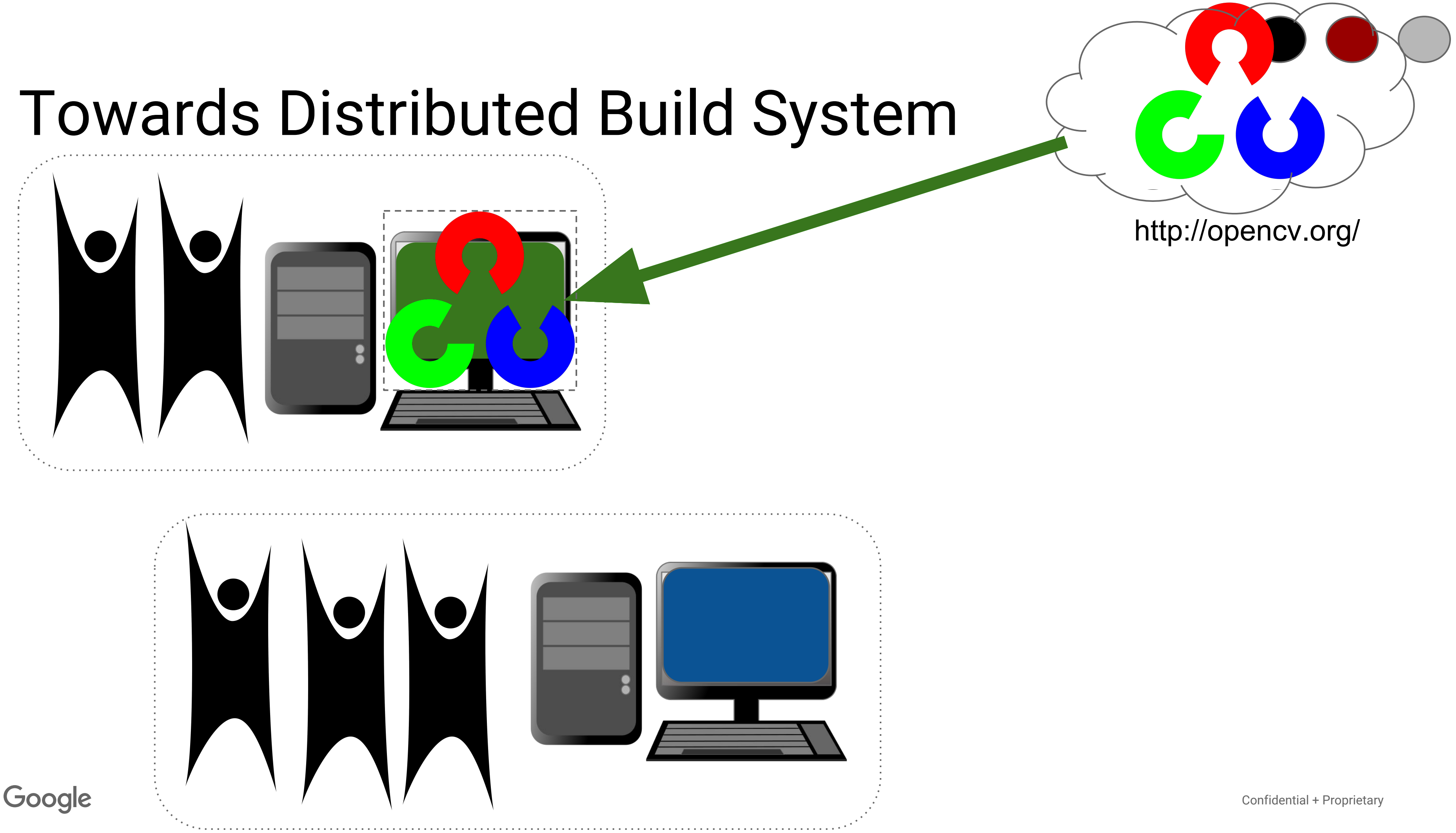
Towards Distributed Build System



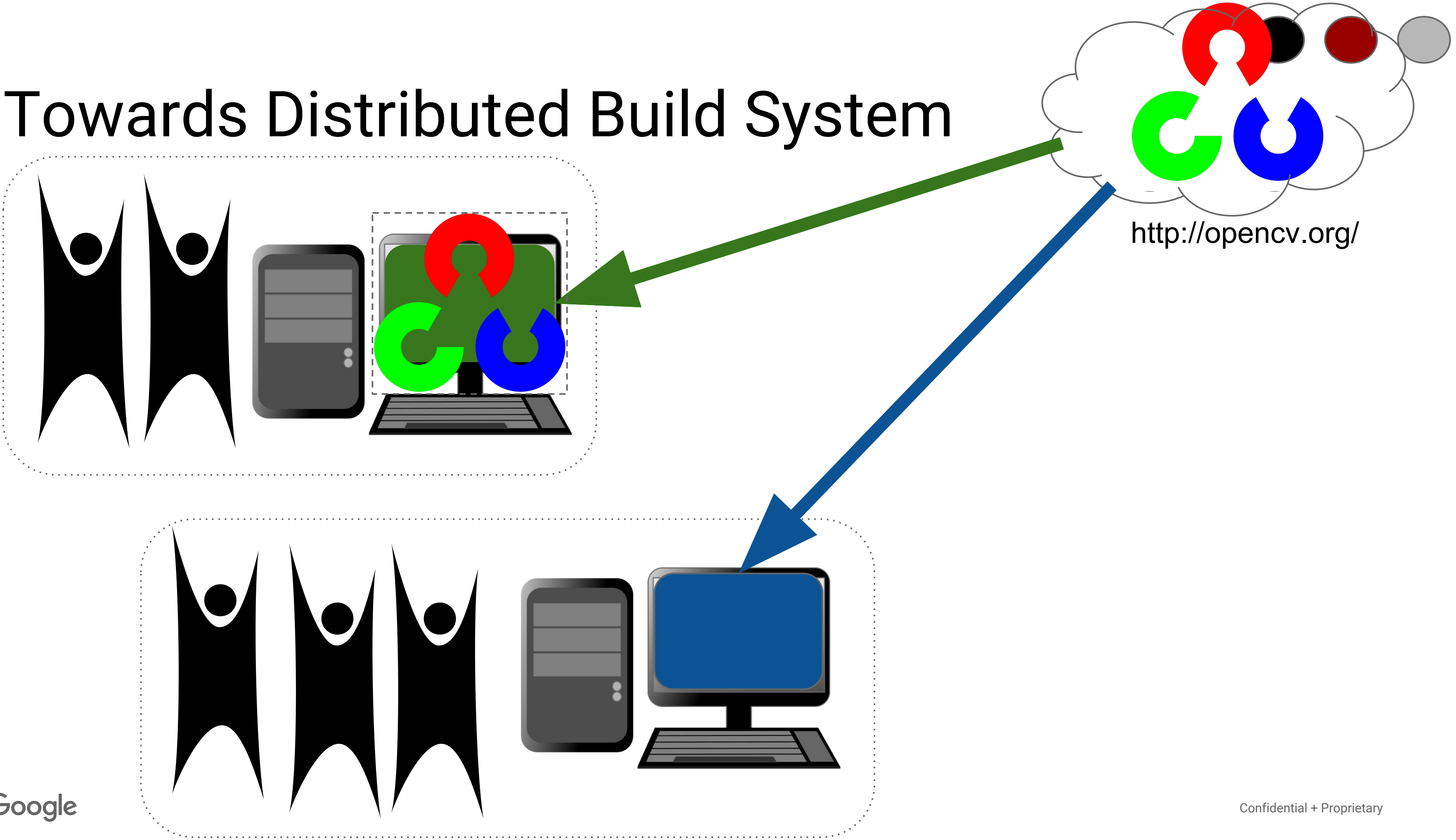
Towards Distributed Build System



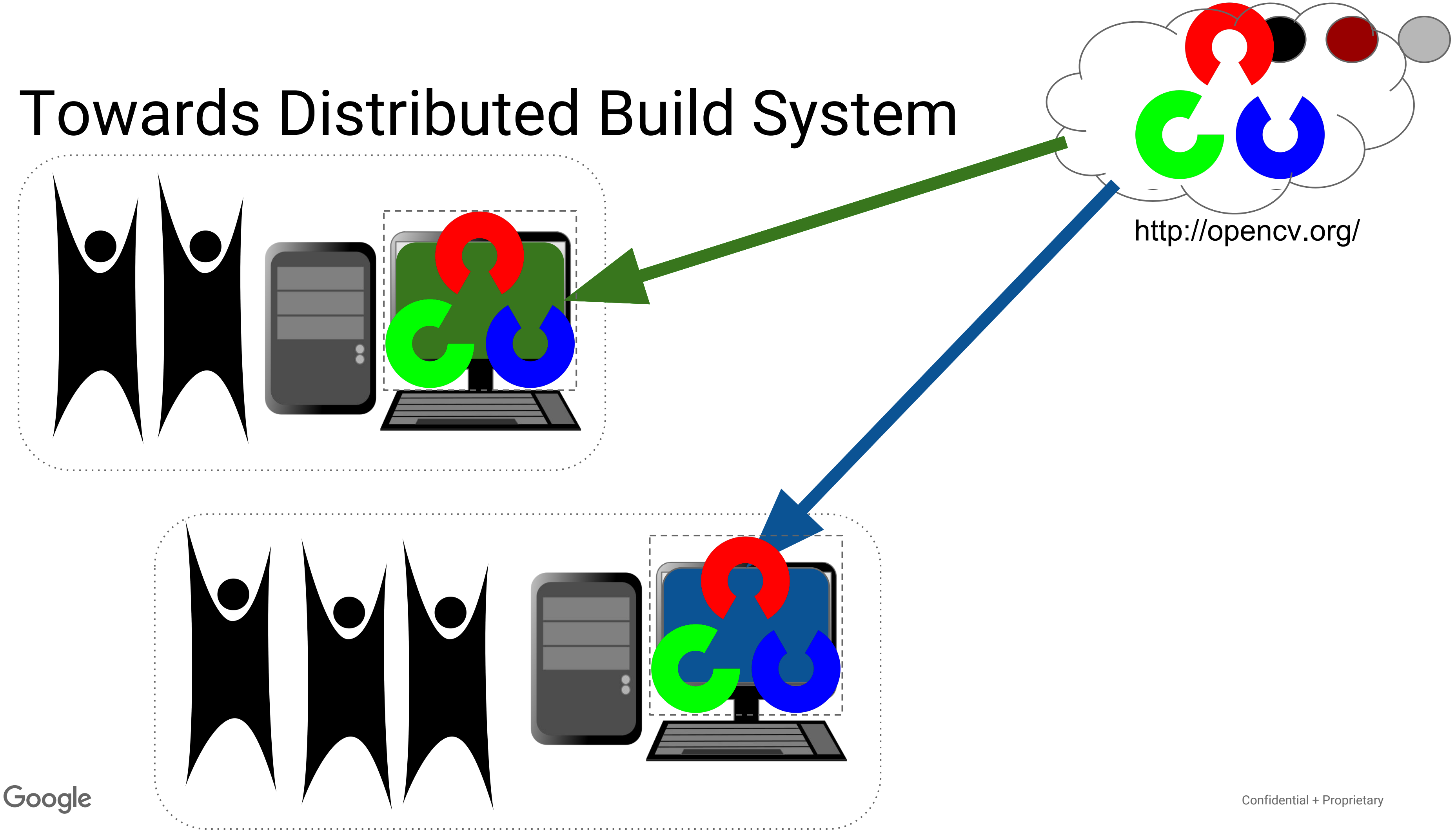
Towards Distributed Build System



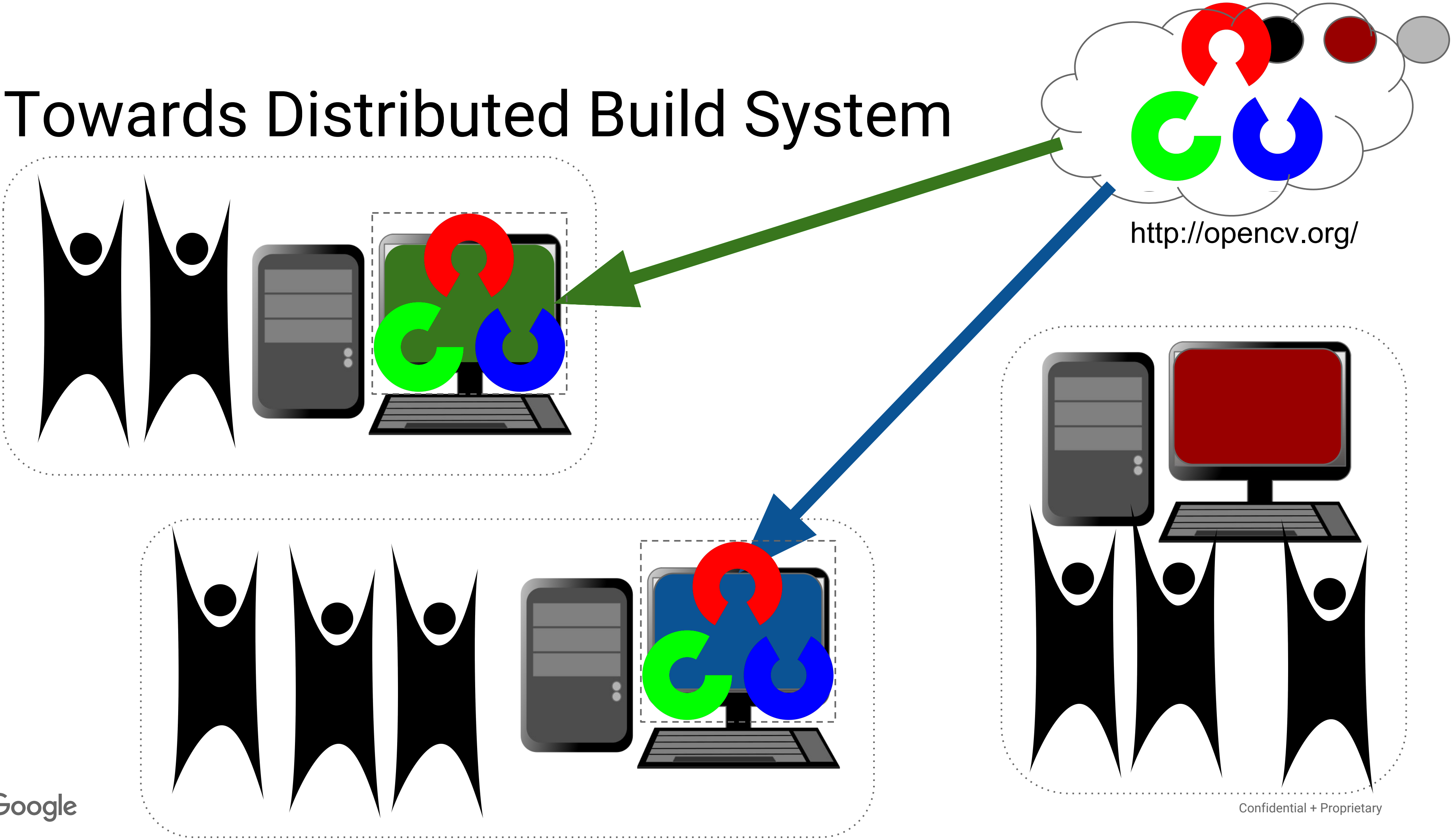
Towards Distributed Build System



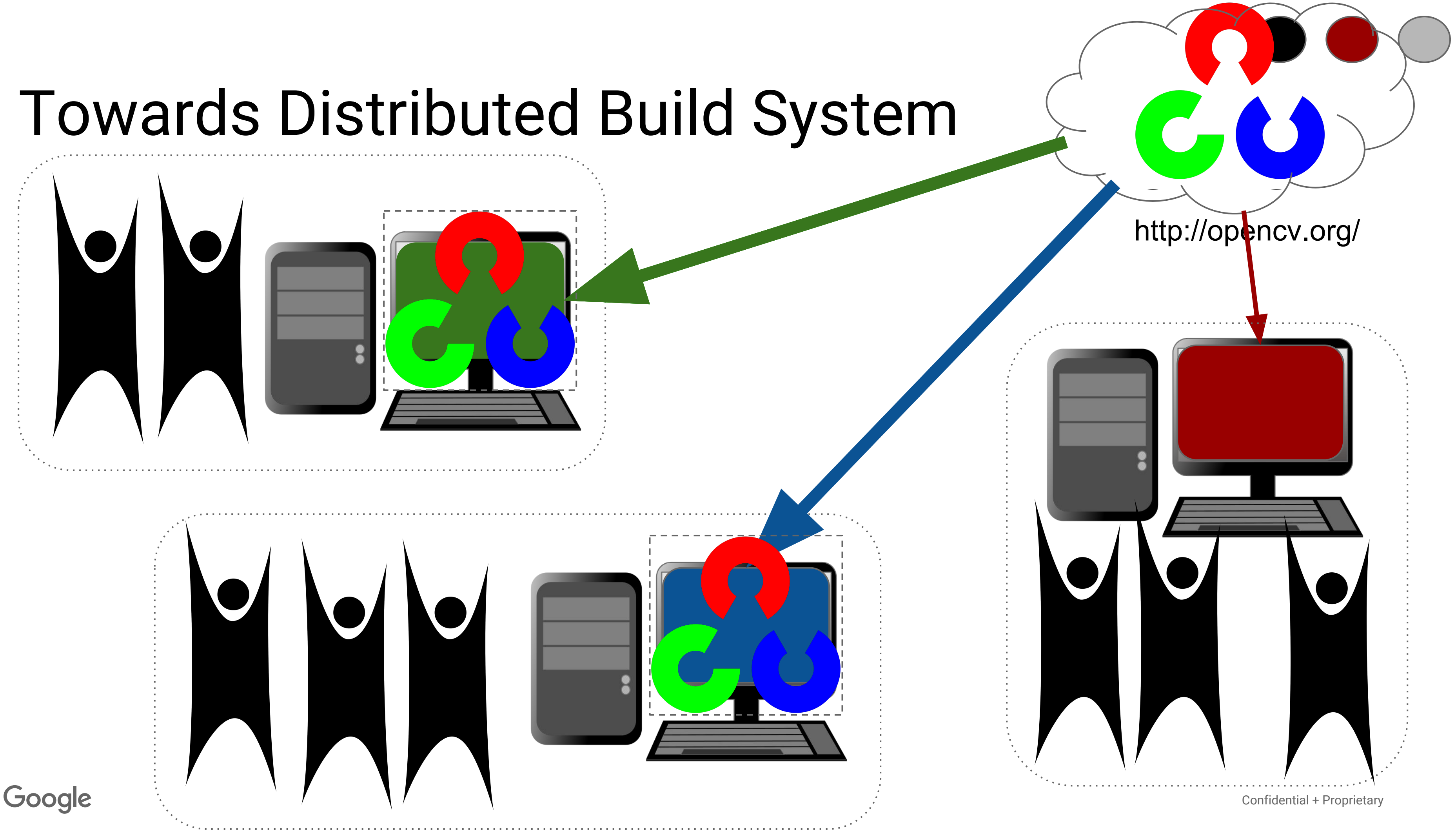
Towards Distributed Build System



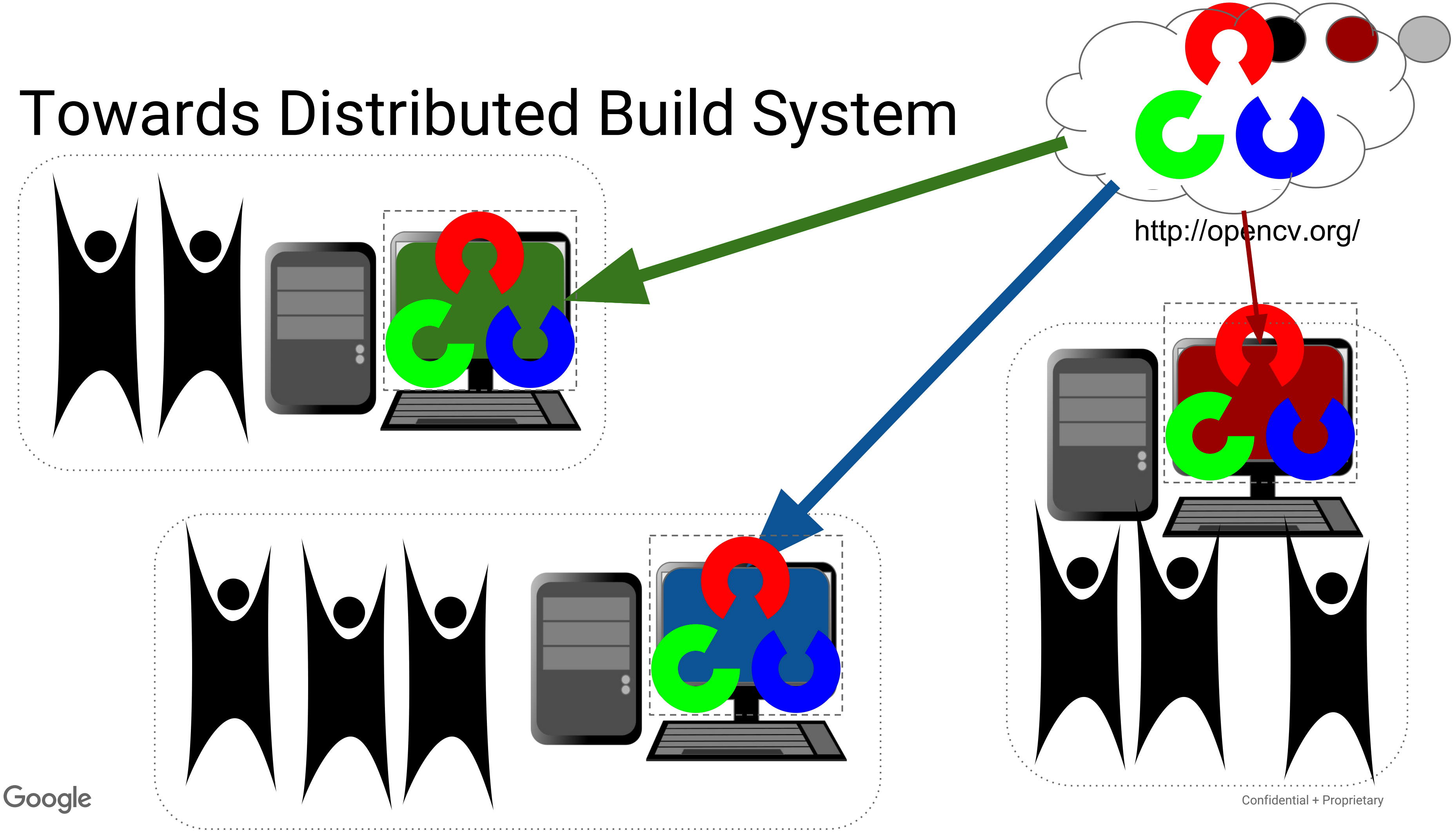
Towards Distributed Build System



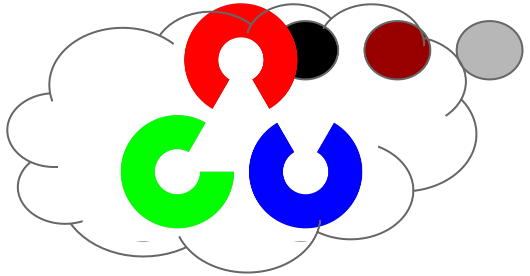
Towards Distributed Build System



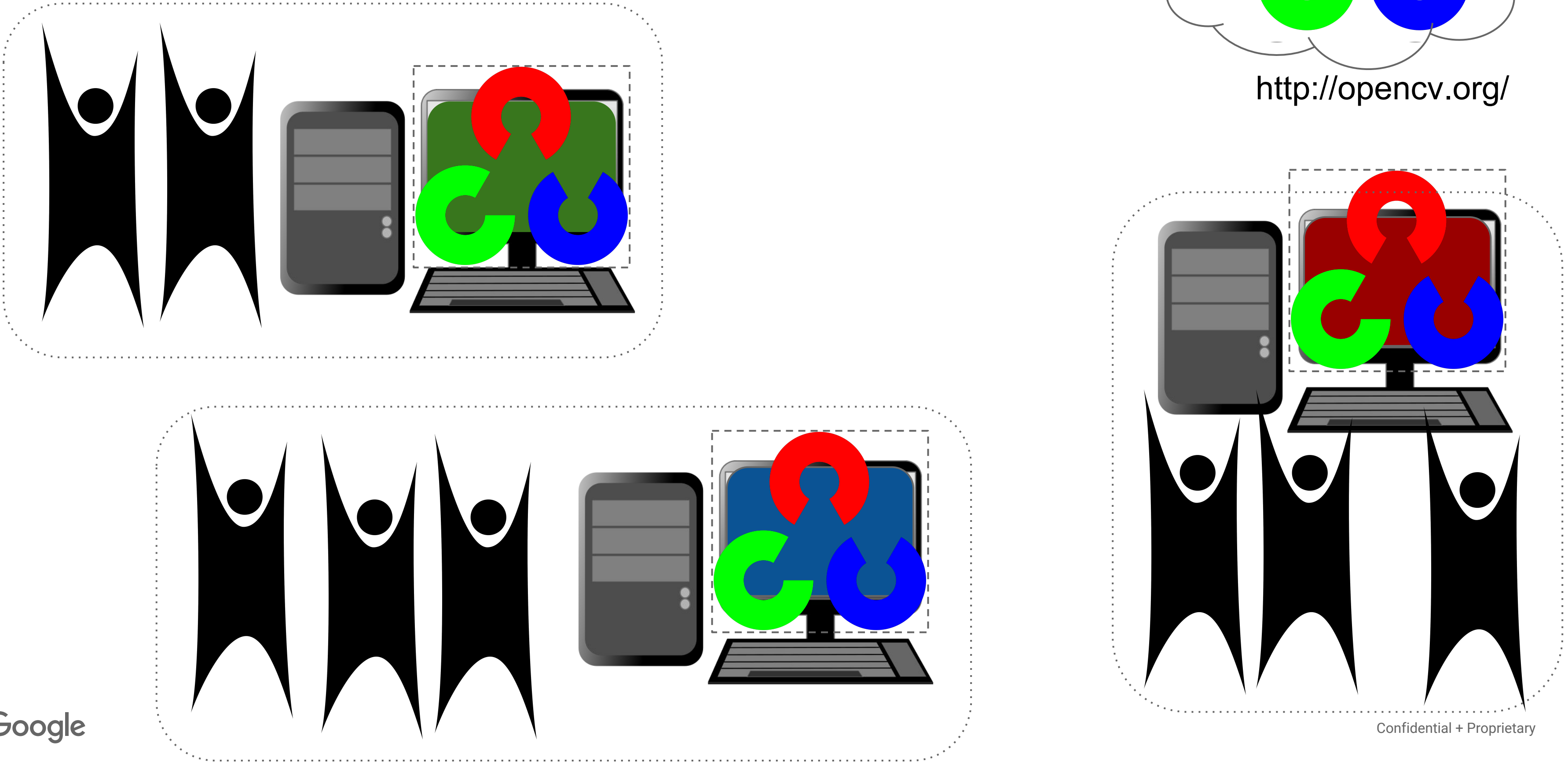
Towards Distributed Build System



Towards Distributed Build System

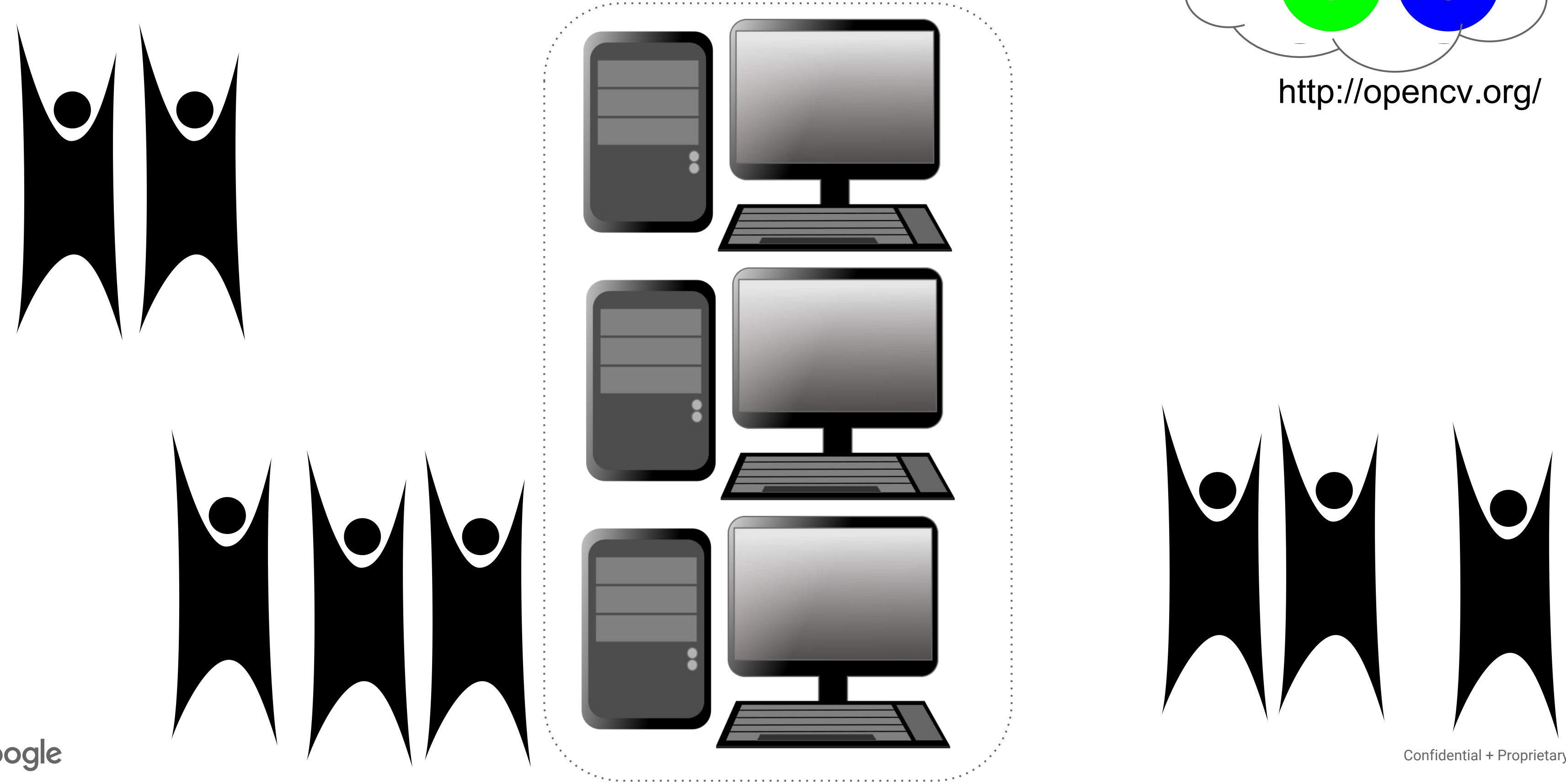


http://opencv.org/



Towards Distributed Build System

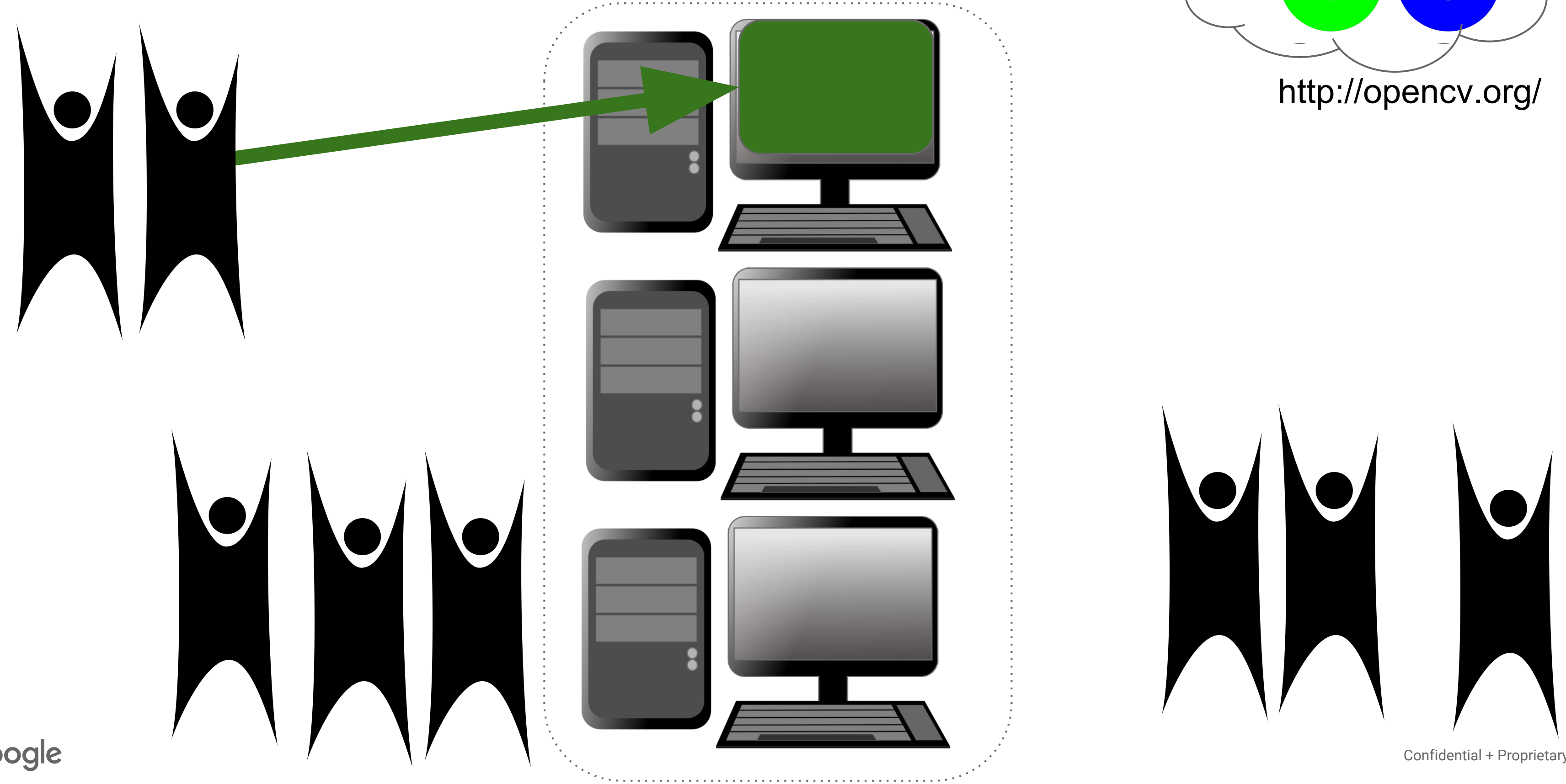
http://opencv.org/



Google

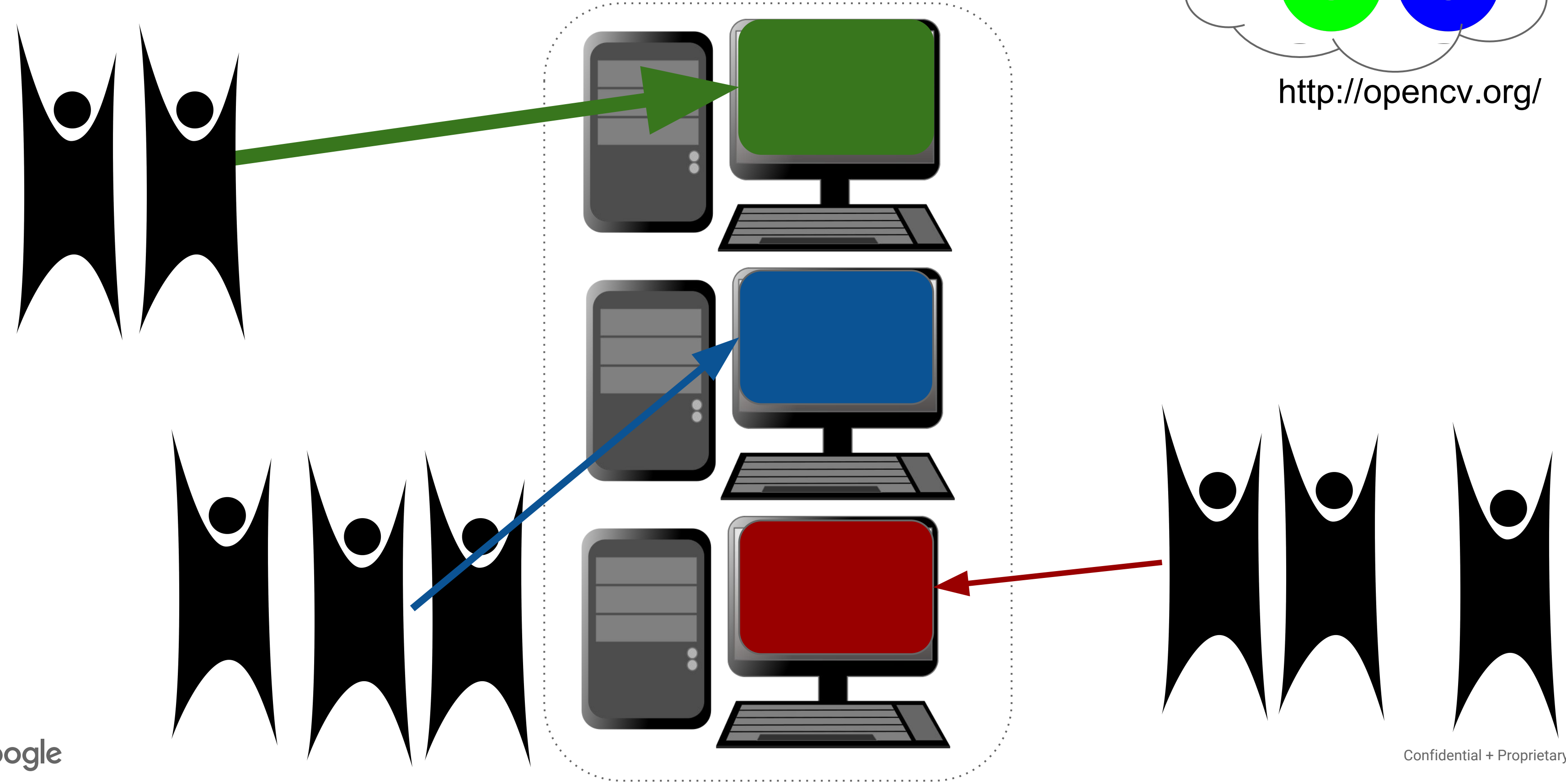
Towards Distributed Build System

http://opencv.org/



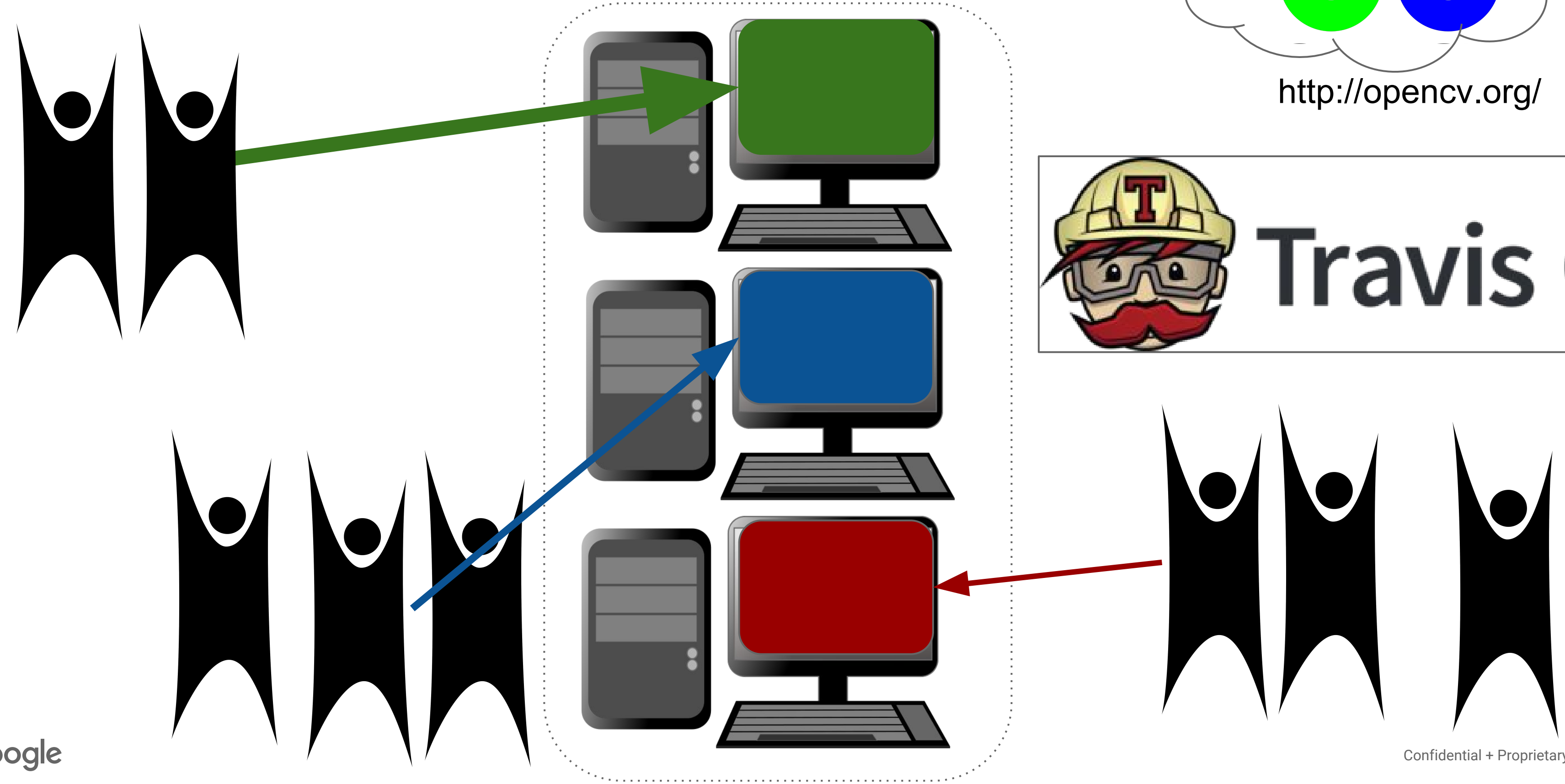
Google

Towards Distributed Build System



http://opencv.org/

Towards Distributed Build System



http://opencv.org/

BuildRabbit: Distributed Build System

BUILD
DEVELOPER INFRASTRUCTURE

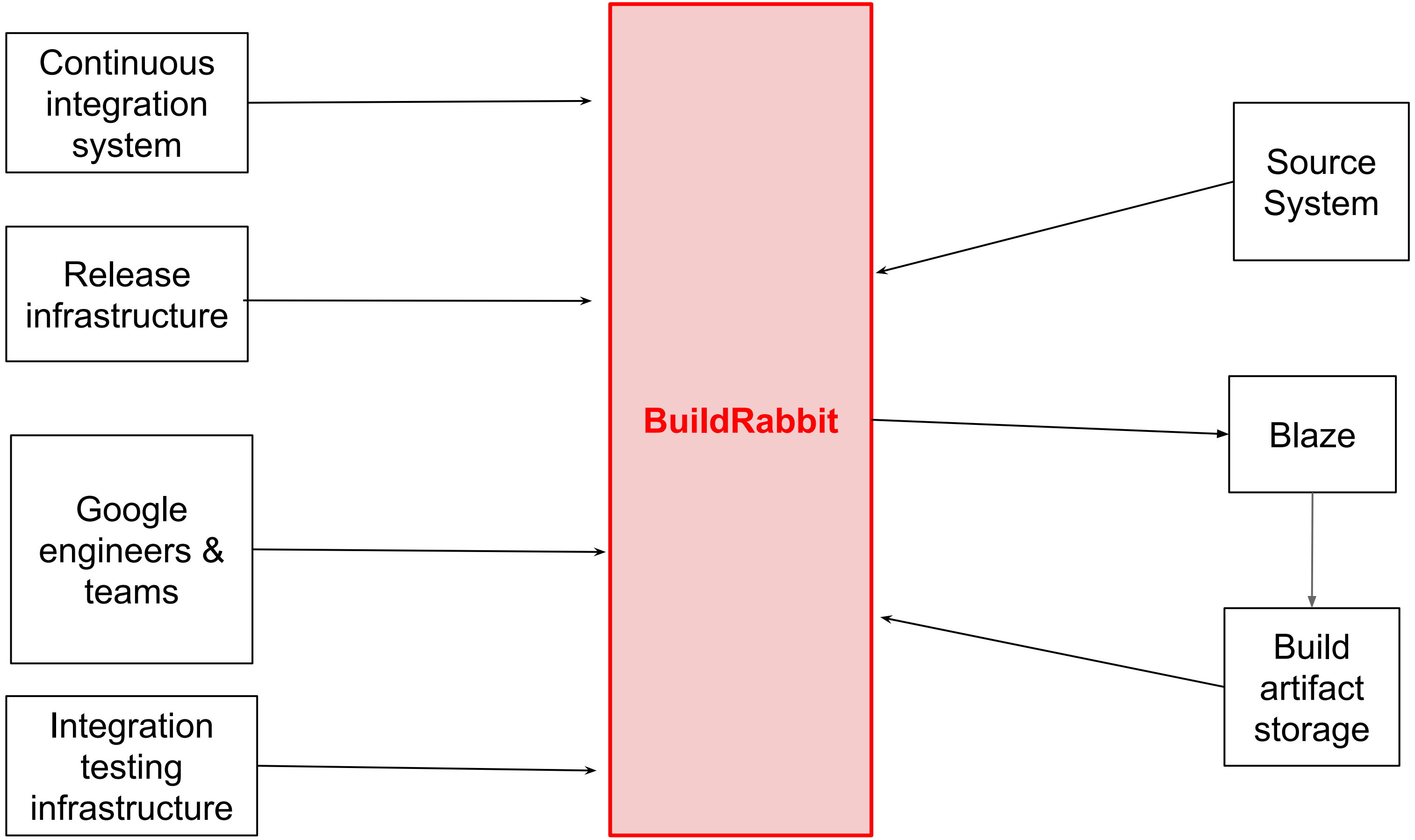
BuildRabbit: Distributed Build System

Bazel

{Fast, Correct} - Choose two

IN THE CLOUD

What does BuildRabbit do?


Continuous integration system
Release infrastructure
Google engineers & teams
Integration testing infrastructure
BuildRabbit
Source System
Blaze
Build artifact storage

Building Distributed Build System at Google Scale

Building
Distributed Build System at Google Scale

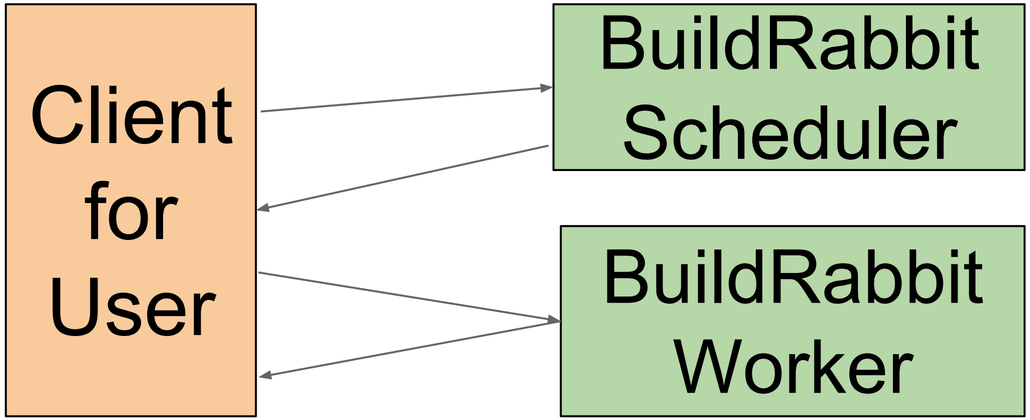
Evolution of BuildRabbit: From push to pull

Evolution of BuildRabbit: From Push to Pull

- Experimental project -> key piece of Google's developer infrastructure

Evolution of BuildRabbit: From Push to Pull

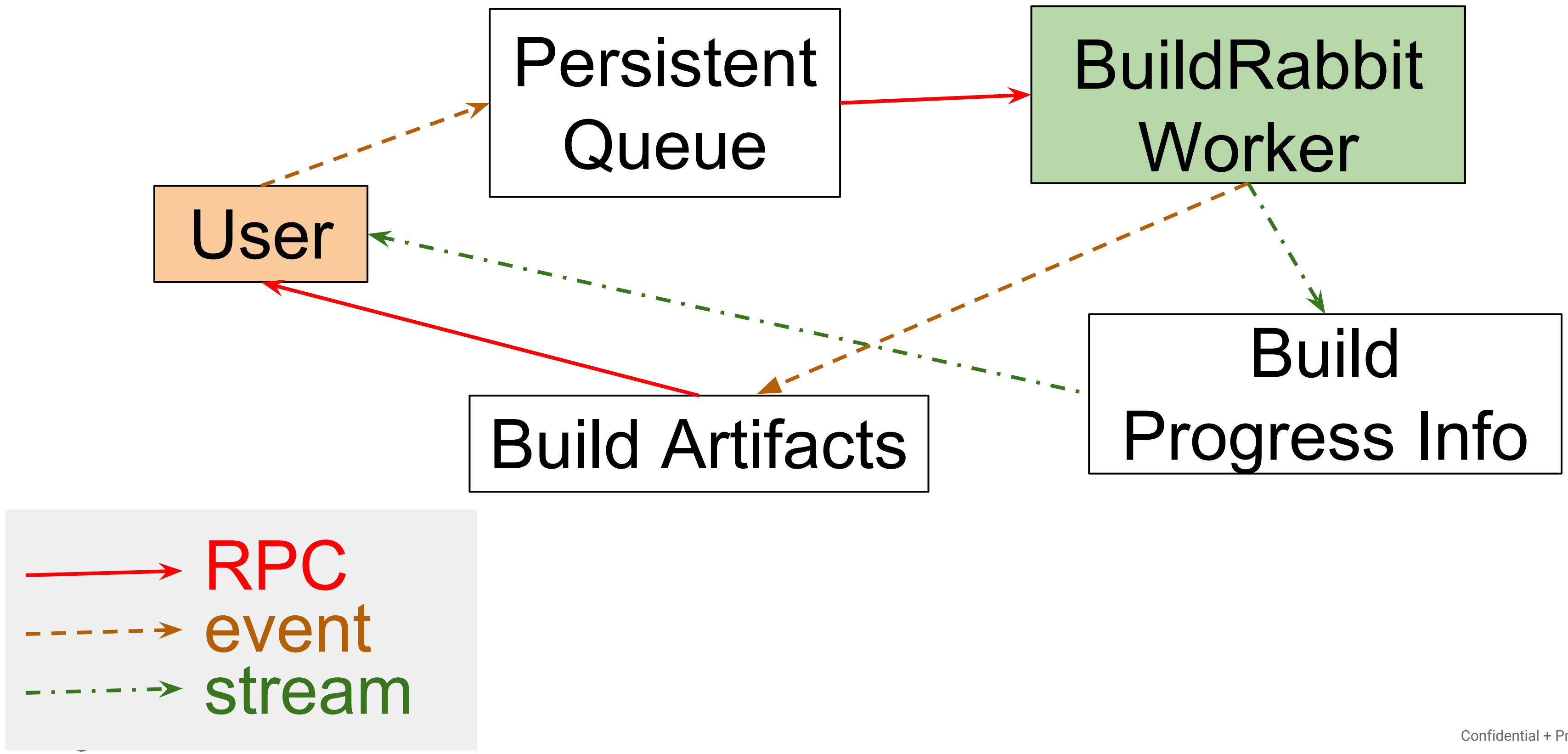
- Experimental project -> key piece of Google's developer infrastructure
- Push model



Evolution of BuildRabbit: From Push to Pull

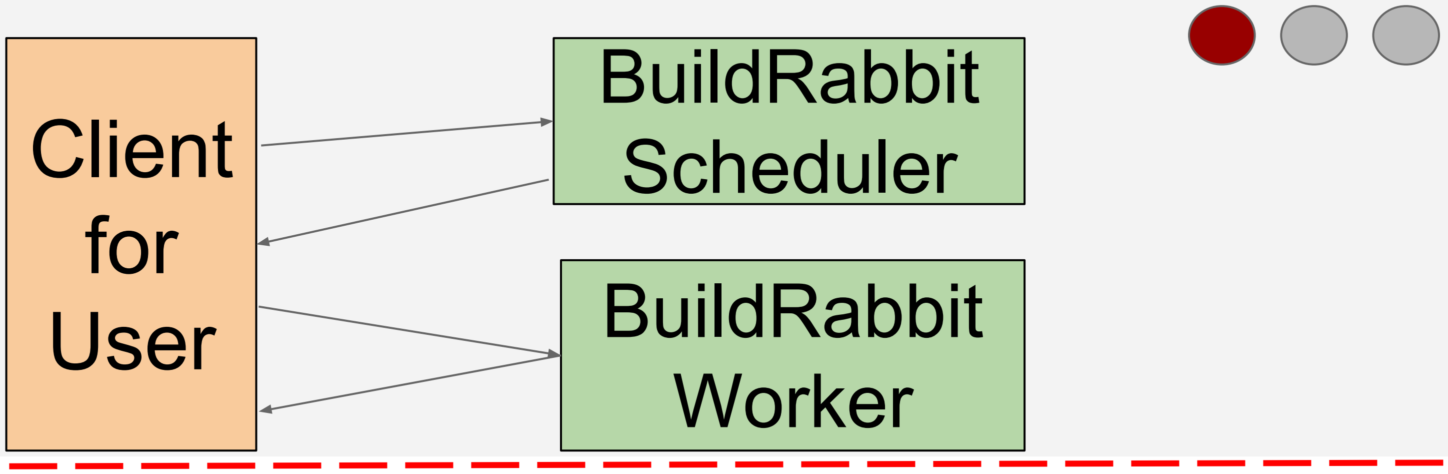
- Experimental project -> key piece of Google's developer infrastructure
- Push model
- Pull model: build service

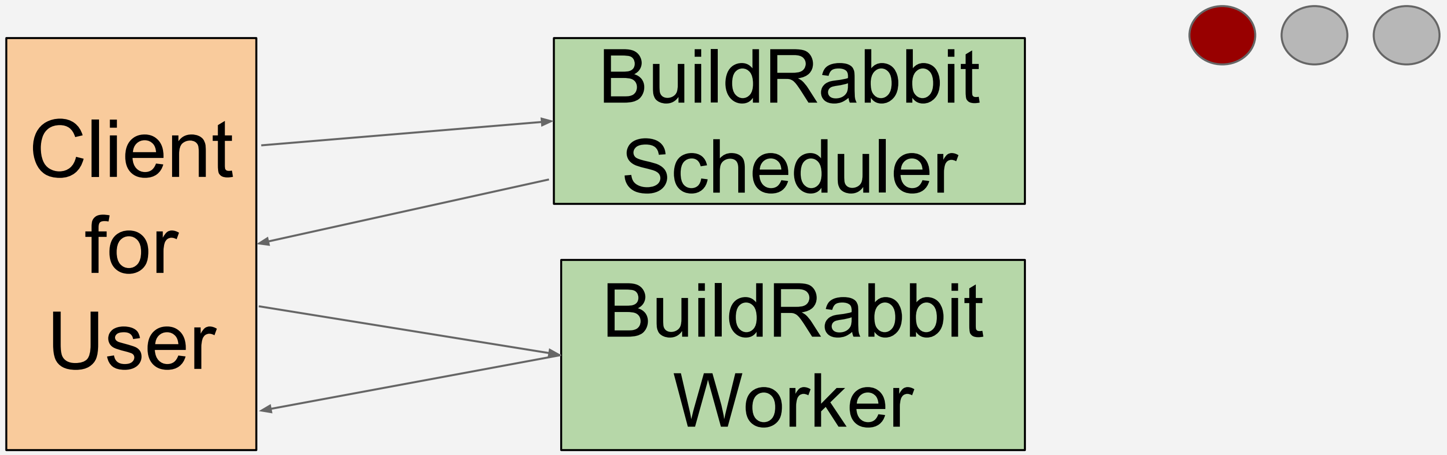
Build Service

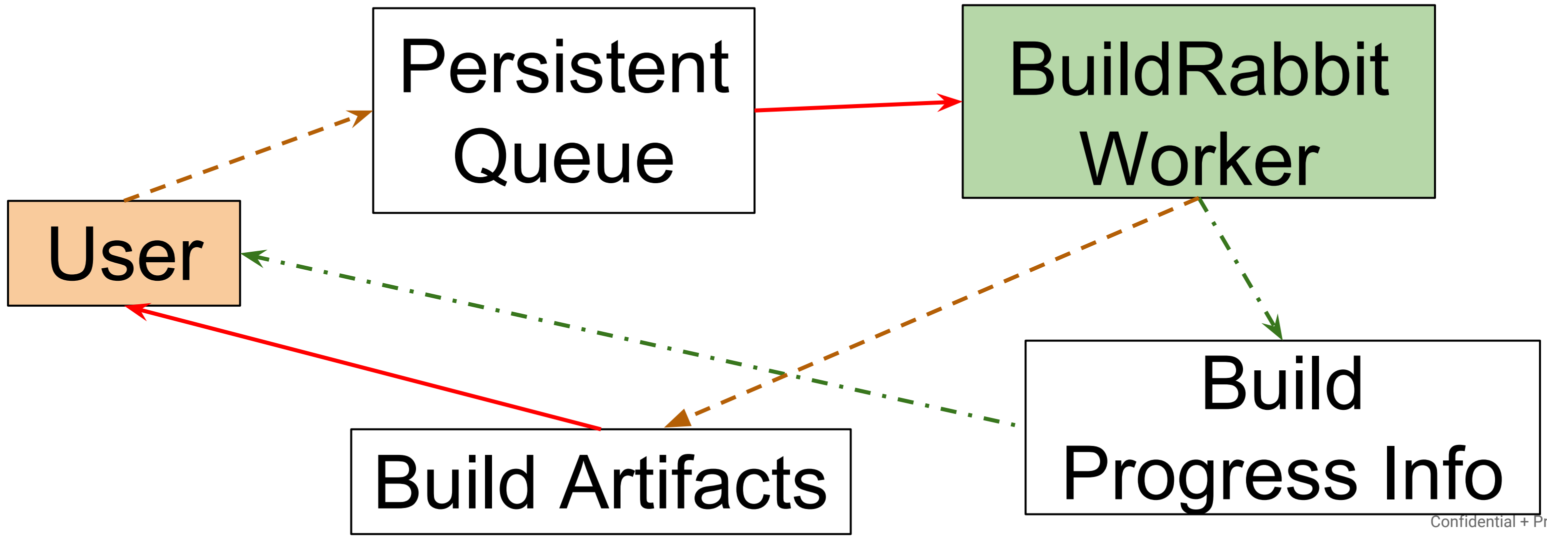


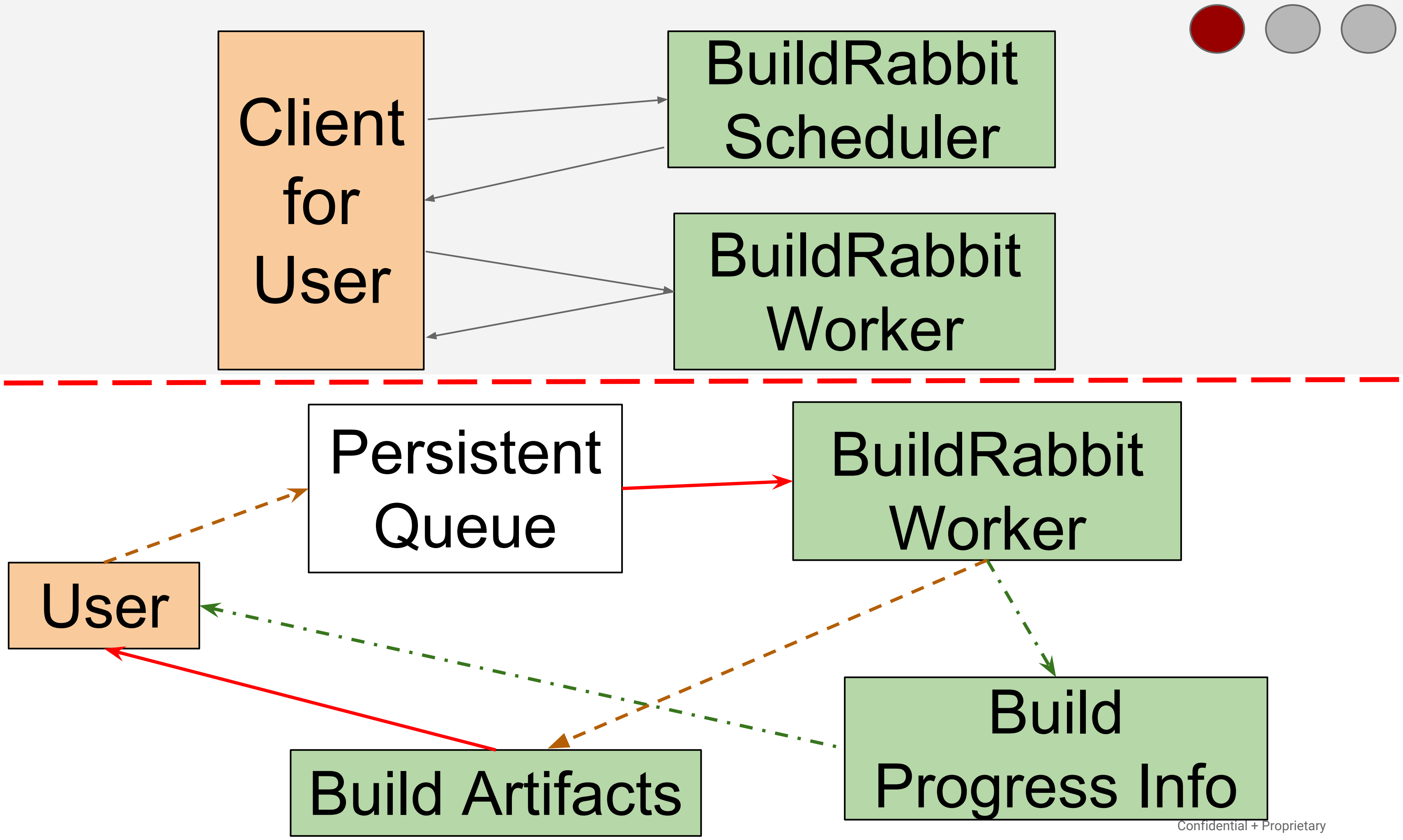
Replacing Jet Engine In-Flight

Re n-


Client for User
BuildRabbit Scheduler
BuildRabbit Worker


Client for User
BuildRabbit Scheduler
BuildRabbit Worker


Persistent Queue
BuildRabbit Worker
User
Build Progress Info
Build Artifacts


Client for User
BuildRabbit Scheduler
BuildRabbit Worker
Persistent Queue
BuildRabbit Worker
User
Build Progress Info
Build Artifacts

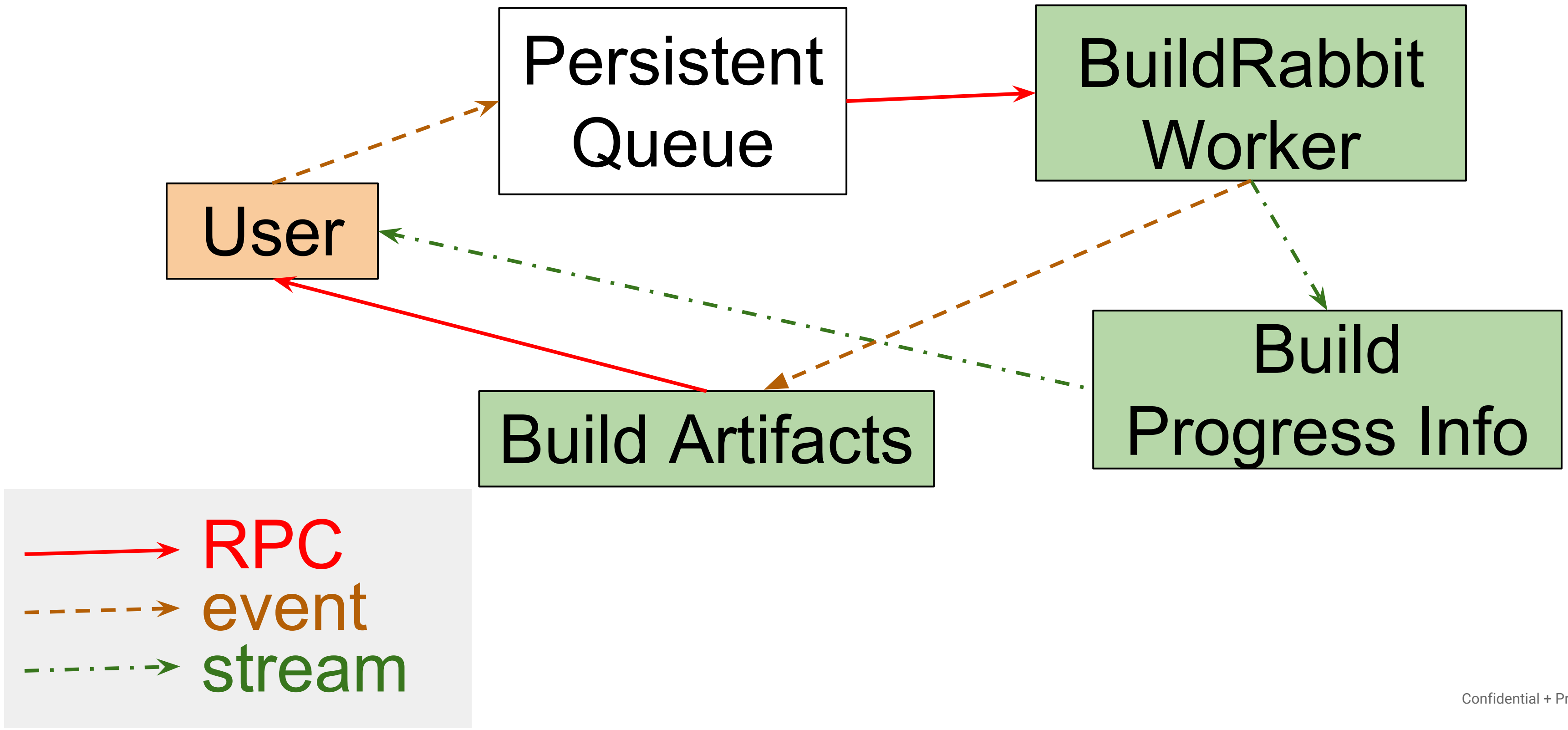
Implementing Re-architected Build System

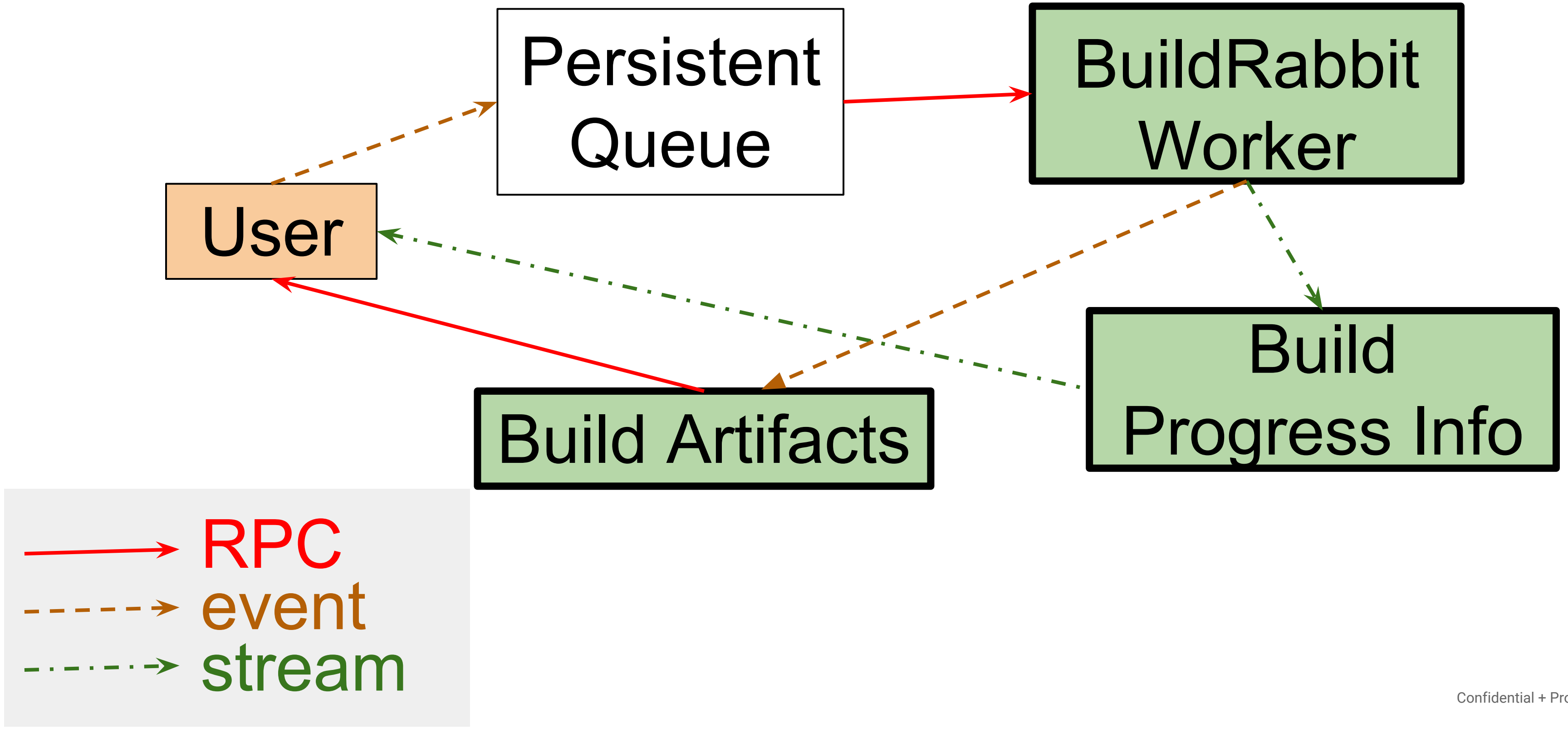
with zero downtime

Migrate backends first

implementing re-architected build system

with zero downtime


Persistent Queue
BuildRabbit Worker
User
Build Artifacts
Build Progress Info
RPC
event
stream


Persistent Queue
BuildRabbit Worker
User
Build Progress Info
Build Artifacts
RPC
event
stream

●○○

Throw-away code is needed to prove

identical functionality

implementing re-architected build system

with zero downtime

● ● ●

Throw-away code is needed to prove

identical functionality

>>> Compatibility window is several months

implementing re-architected build system

with zero downtime

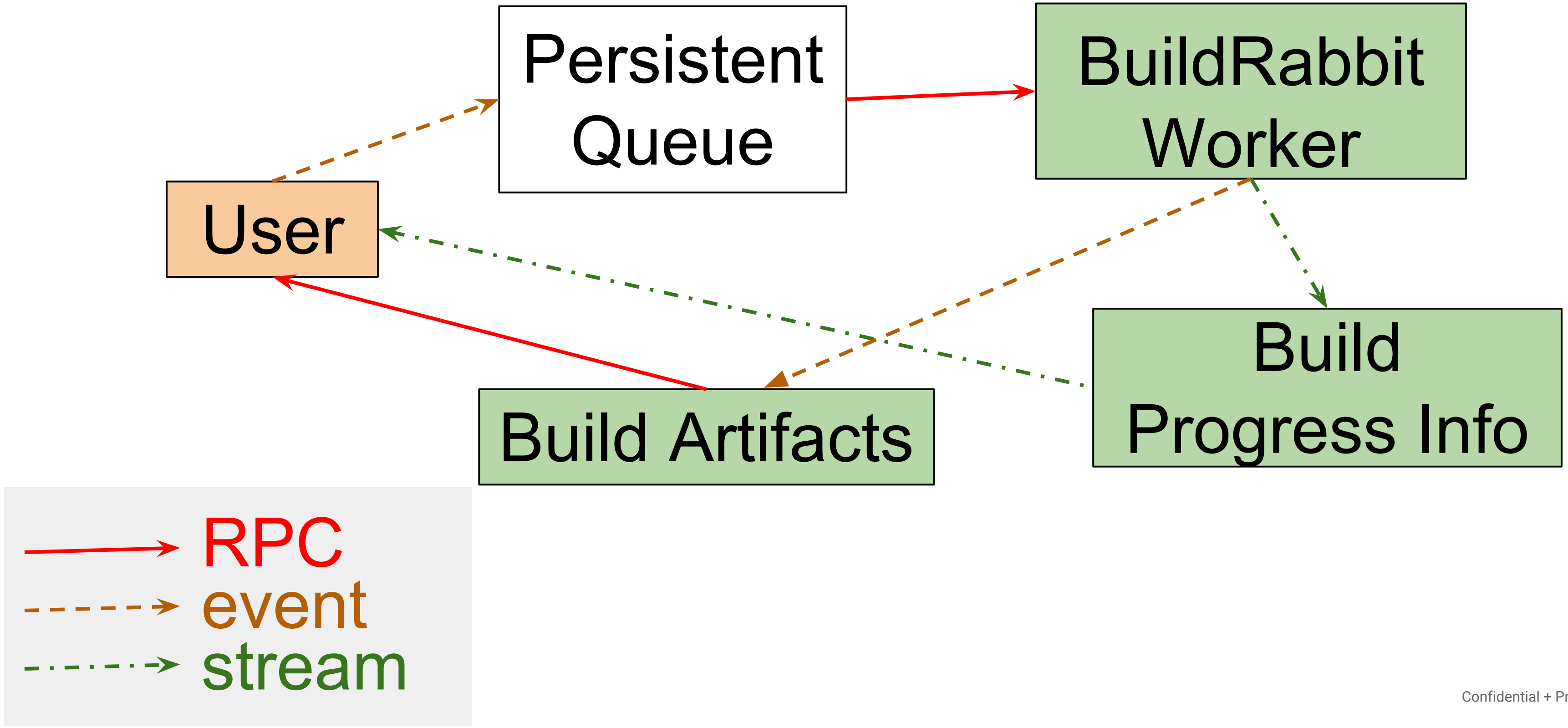
●○○

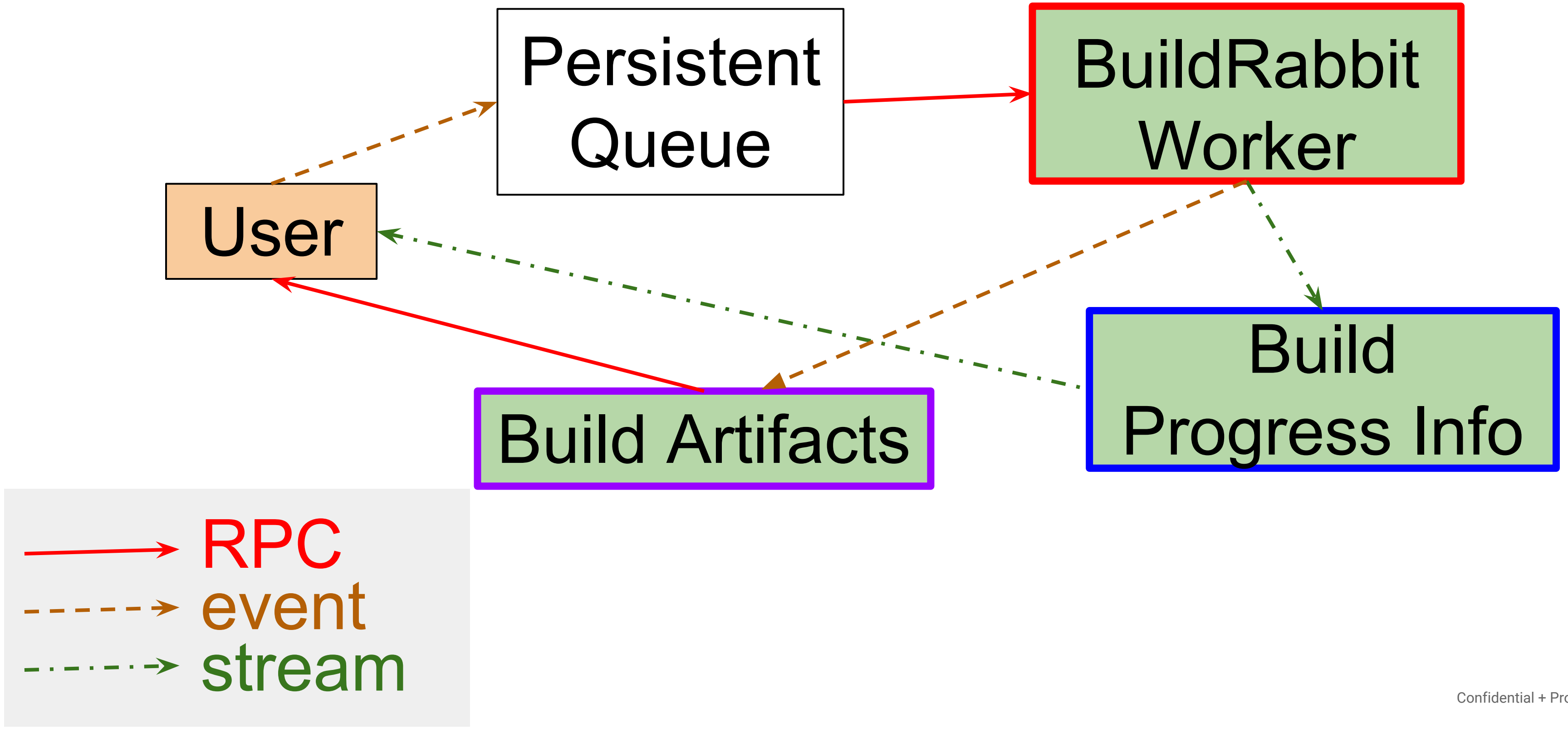
Target launch-friendly clients

before transitioning everyone

implementing re-architected build system

with zero downtime

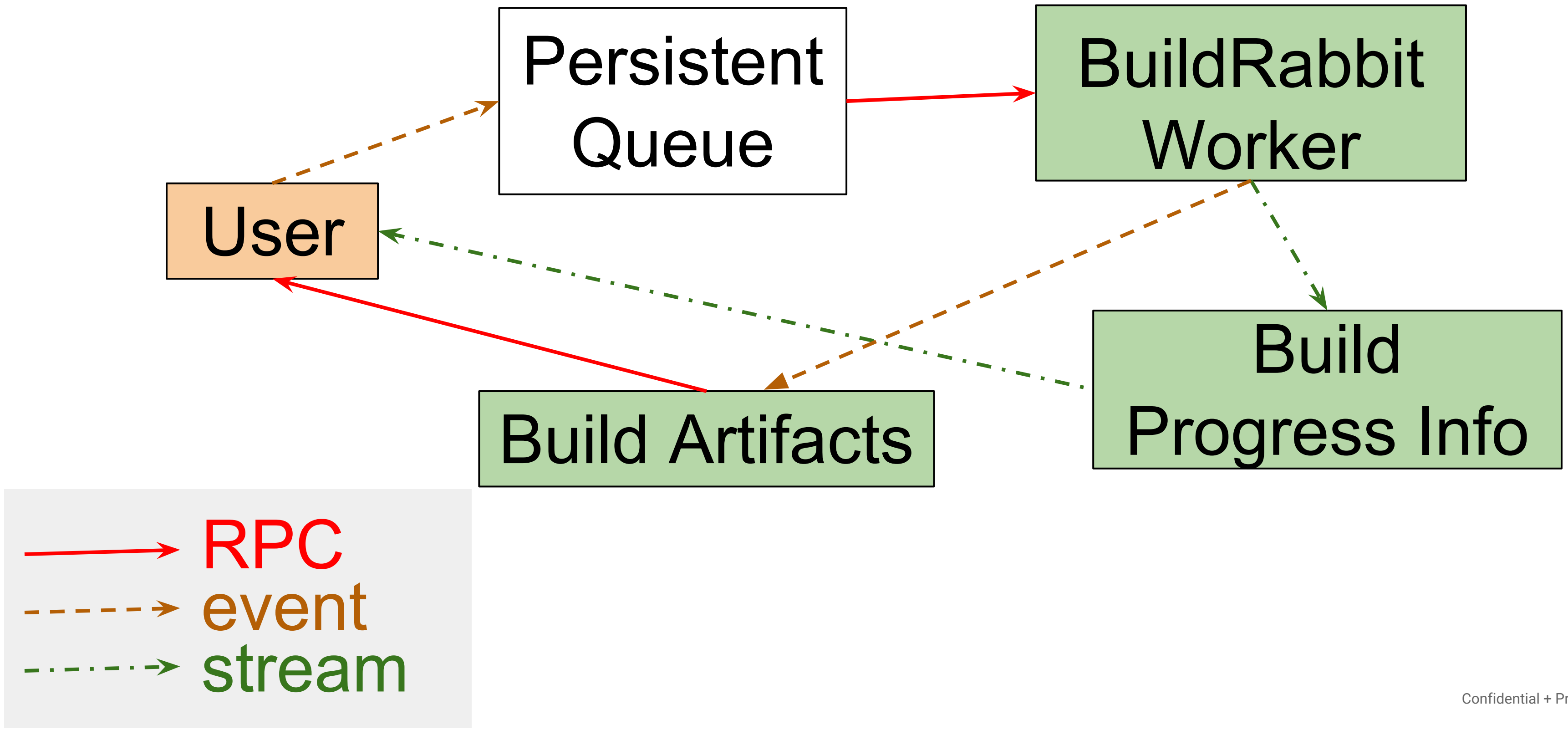

Persistent Queue
BuildRabbit Worker
User
Build Progress Info
Build Artifacts
RPC
event
stream

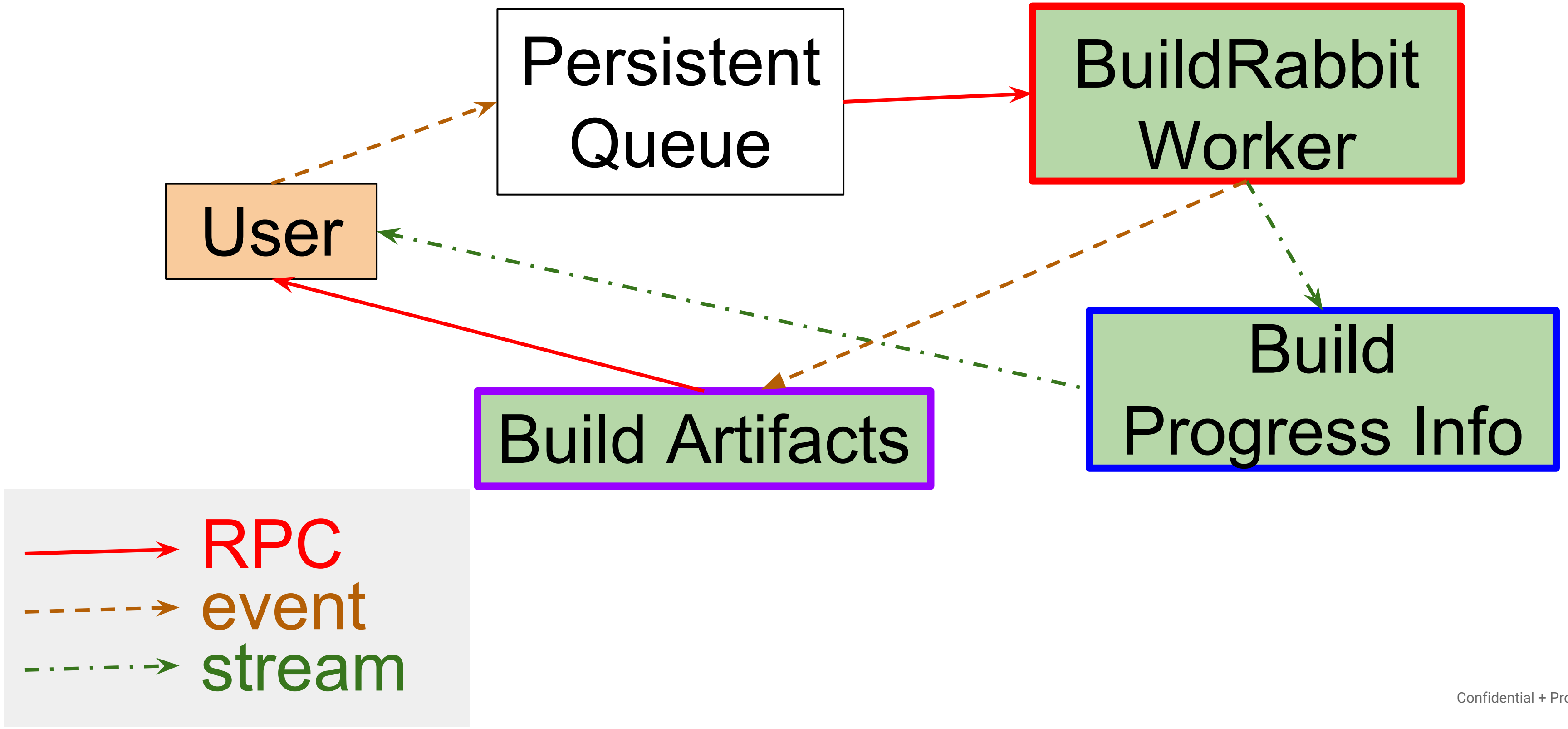

Persistent Queue
BuildRabbit Worker
User
Build Artifacts
Build Progress Info
RPC
event
stream

Decouple launch of services

implementing re-architected build system

with zero downtime


Persistent Queue
BuildRabbit Worker
User
Build Artifacts
Build Progress Info
RPC
event
stream


Persistent Queue
BuildRabbit Worker
User
Build Progress Info
Build Artifacts
RPC
event
stream

●●●

Maximum visibility into system state

implementing re-architected build system
with zero downtime

● ● ●

Practice before launch

implementing re-architected build system

with zero downtime

Practice before launch, a lot

implementing re-architected build system

with zero downtime

Practice before launch, a lot

Have a solid rollback plan

implementing re-architected build system
with zero downtime

References

- Why Google Stores Billions of Lines of Code in a Single Repository https://www.youtube.com/watch?v=W71BTkUbdqE
- Build artifacts storage http://google-engtools.blogspot.com/2011/10/build-in-cloud-distributing-build.html
- Distributed build actions http://google-engtools.blogspot.com/2011/09/build-in-cloud-distributing-build-steps.html
- Continuous integration testing http://dl.acm.org/citation.cfm?id=2635910

Gratitude

Vince Noel

Scott Zawalski

Rob Siemborski

BuildRabbit team

Caitie McCaffrey

David Greenberg

Google

goto; conference

Please

Remember to rate this session

Thank you!