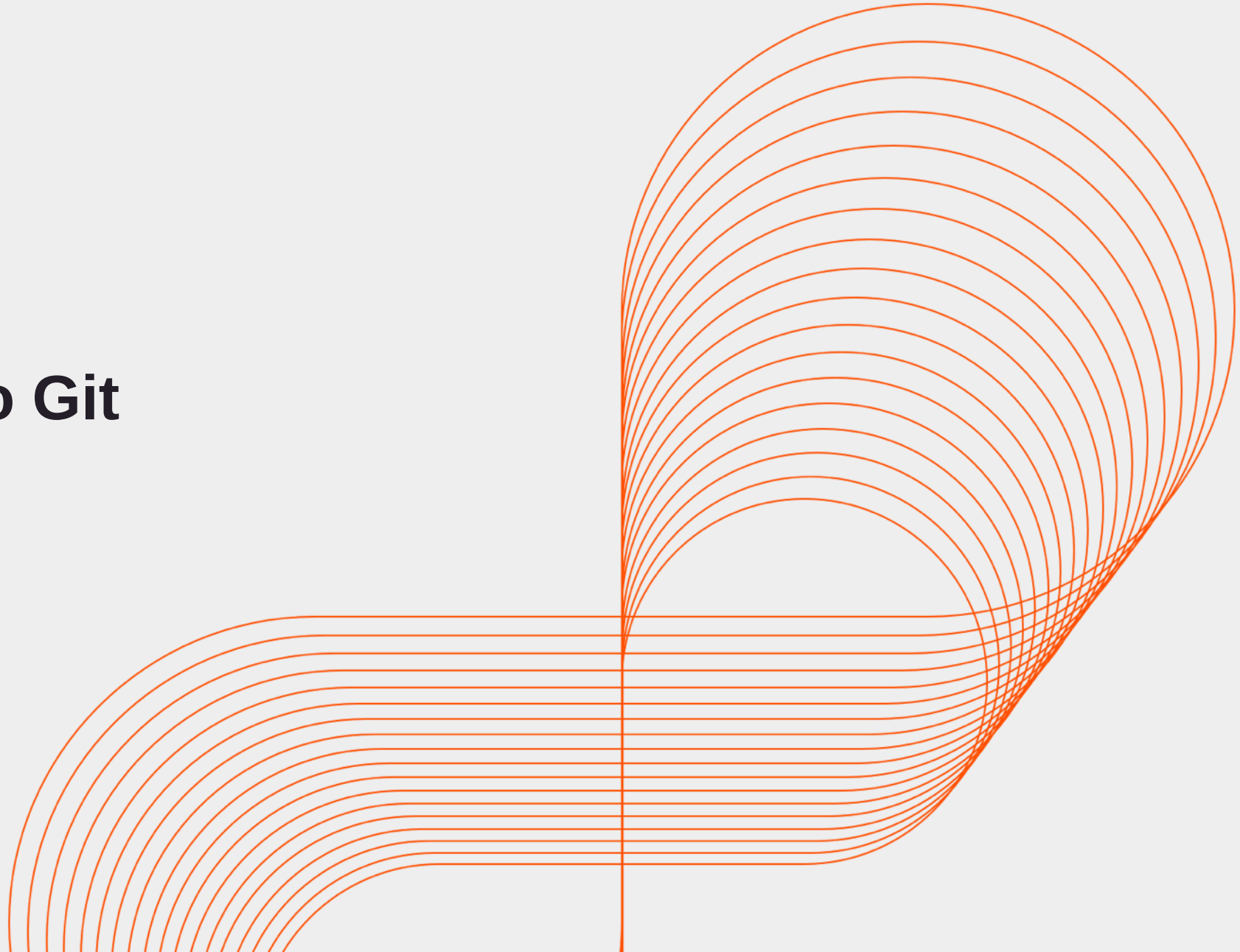
Persistent

Git:

Introduction to Git

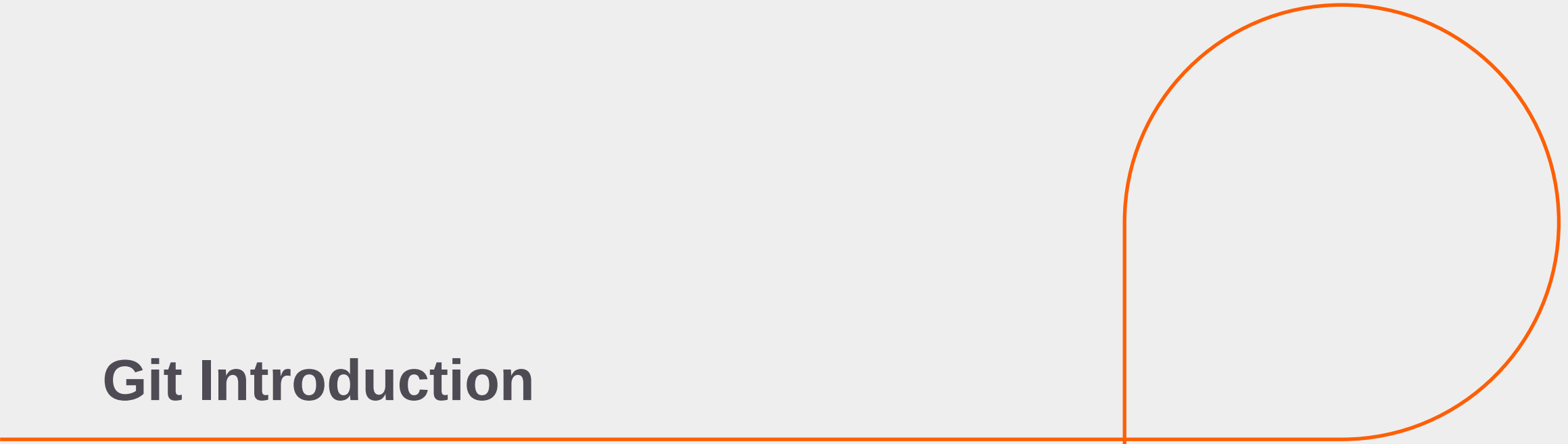
Persistent University



Key Learning Points

- What is VCS?
- What is Git?
- Centralized vs Distributed VCS
- Installing Git on Windows
- Configuring Git

Git Introduction



DevOps is:

The practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

Characterized by operations staff making use many of the same techniques as developers for their systems work.

A culture of trust and collaboration in which people use right Tools for automation to achieve Continuous Delivery and Continuous Deployments.

Project Execution Methodologies

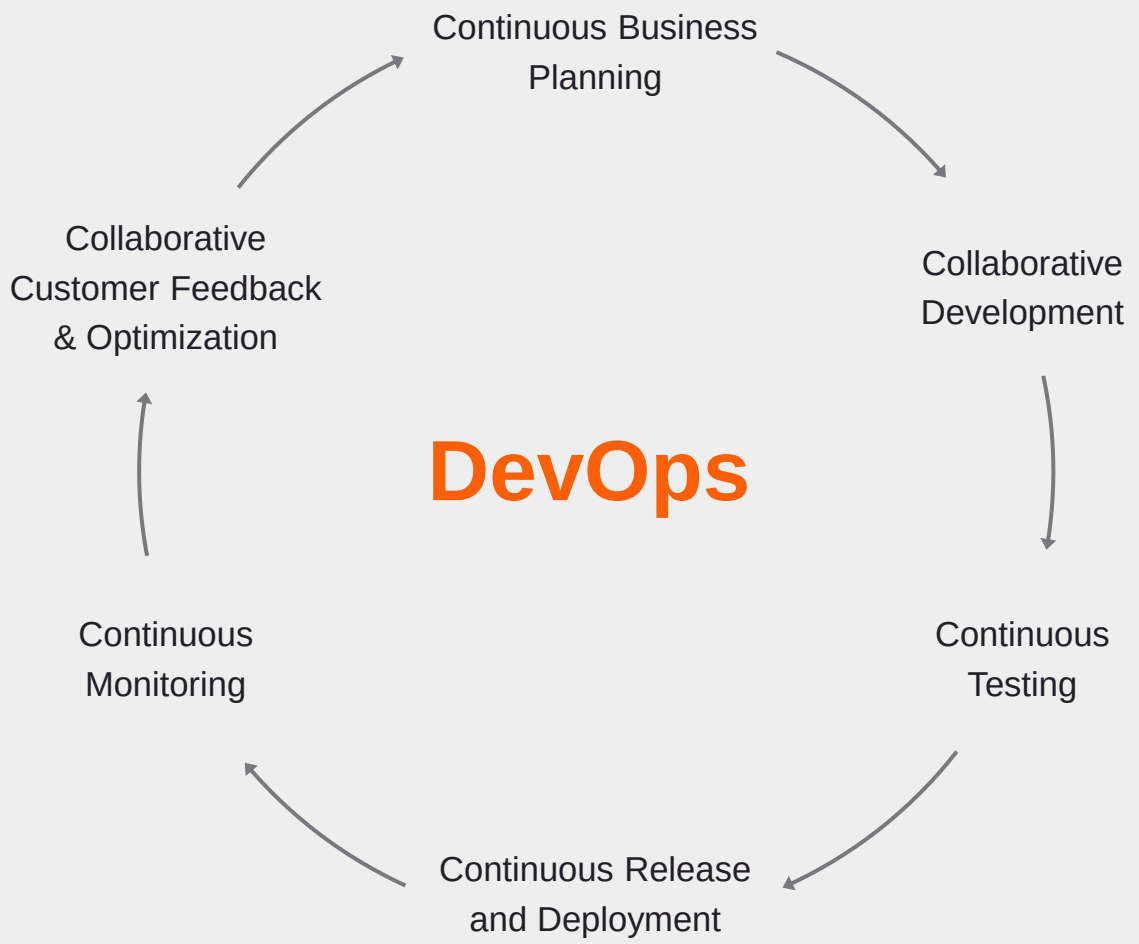
Agile Waterfall

Waterfall

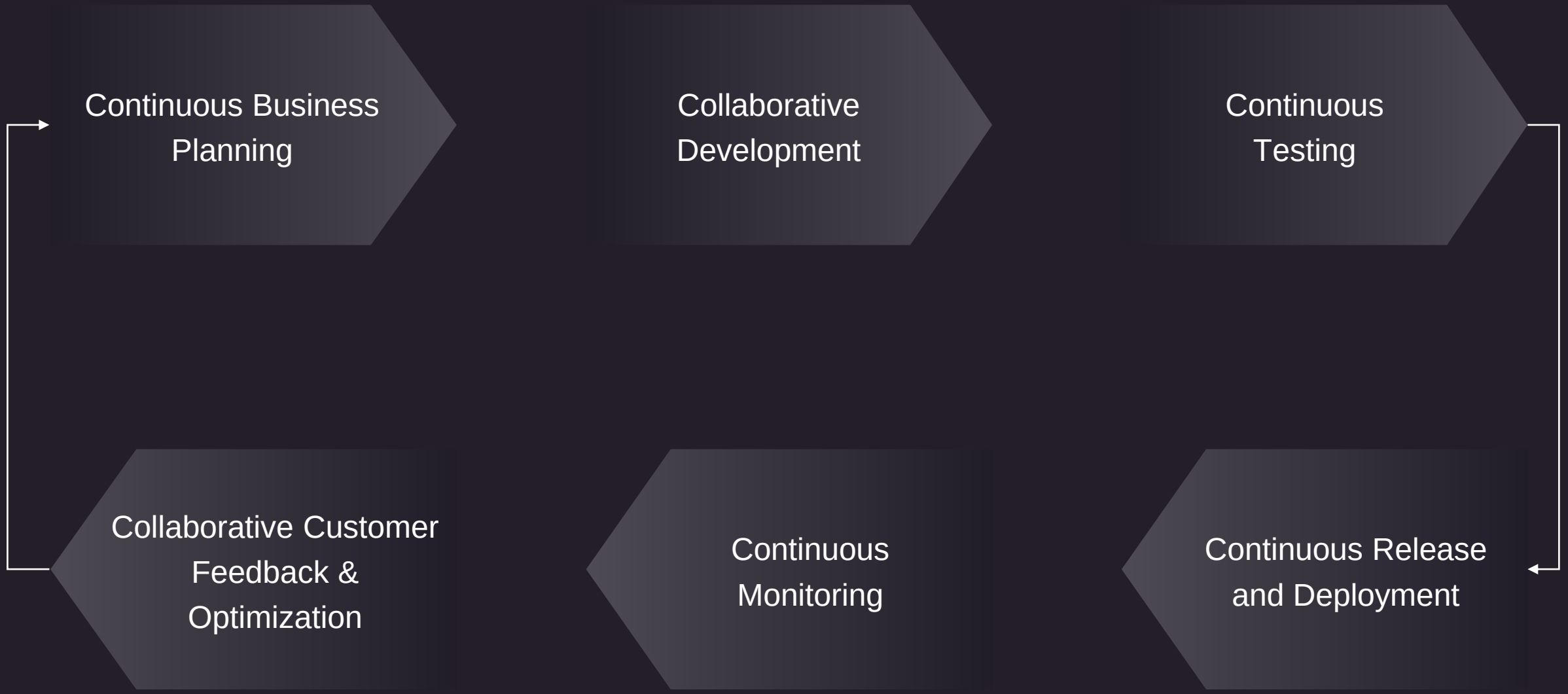
Agile

DevOps

DevOps Lifecycle



DevOps Lifecycle



DevOps Tools

Development

DevOps Tools Market Map

Source Code Management

git PERFORCE SUBVERSION mercurial

Containers

XebiaLabs Now Digital.ai
ZEROTURNAROUND
Jelastic docker elasticbox

Continuous Integration

Build

<APACHE ANT>
Maven
Gradle

Travis TC MaestroDev circleci
CloudBees CodeShip cruisecontrol Hudson
ATLASSIAN Bamboo
wercker semaphore Jenkins

Configuration Management

puppet labs CHEF SALTSTACK smartfrog glu
CFEngine BCFG2 ScriptRock Juju

Deployment

ANSIBLE
Capistrano deploy urban{code} $ fab
HashiCorp Vagrant Vlad the Deployer FUNC

Testing

cucumber FitNesse JUnit TESTING ANYWHERE TestNG
arquillian Capybara eggplant
Selenium SAUCELABS Progress Telerik

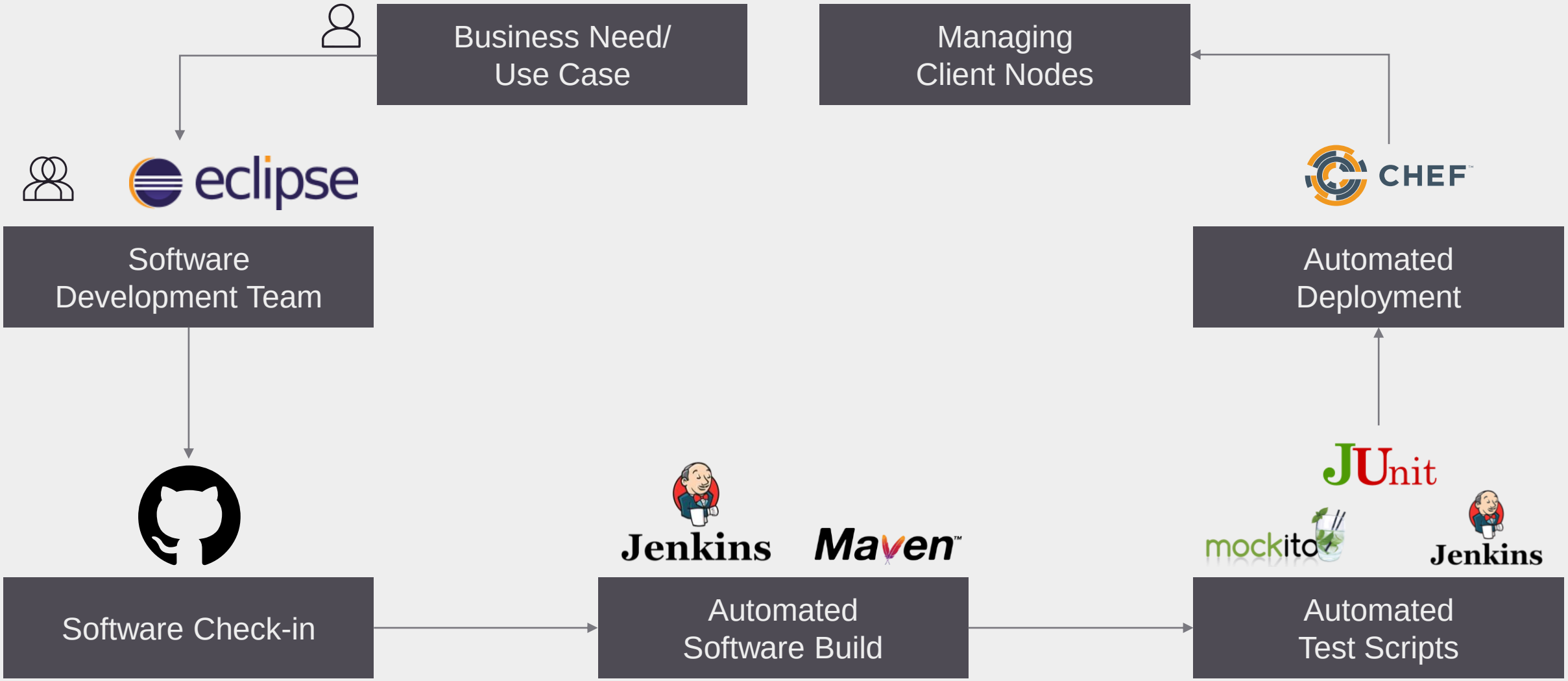
Monitoring

Nagios New Relic Airbrake.io PagerDuty
Sensu app enlight CRITTERCISM solarwinds pingdom librato

Operations

Persistent

Development to Deployment Pipeline: Workflow



Disorganized Project

What is Version Control? (contd.)

Overview

Version control (or **revision control**, or **source control**) is all about managing multiple versions of documents, programs, web sites, etc.

- Almost all “real” projects use some kind of version control.
- Essential for team projects, but also very useful for individual projects.

Version Control Systems

Some well-known version control systems are CVS, Subversion, Mercurial, and Git.

- CVS and Subversion use a “central” repository; users “check out” files, work on them, and “check them in”.
- Mercurial and Git treat all repositories as equal.

Newer Systems

Distributed systems like Mercurial and Git are newer and are gradually replacing centralized systems like CVS and Subversion.

Benefits of Version Control

Basic Functionality:

- Keep track of changes made to files (allows roll-backs)
- Merge the contributions of multiple developers

Benefits:

- Facilitates backups
- Increased productivity (vs manual version control)
- Encourages experimentation
- Helps to identify/fix conflicts
- Makes source readily available – Less duplicated effort

History

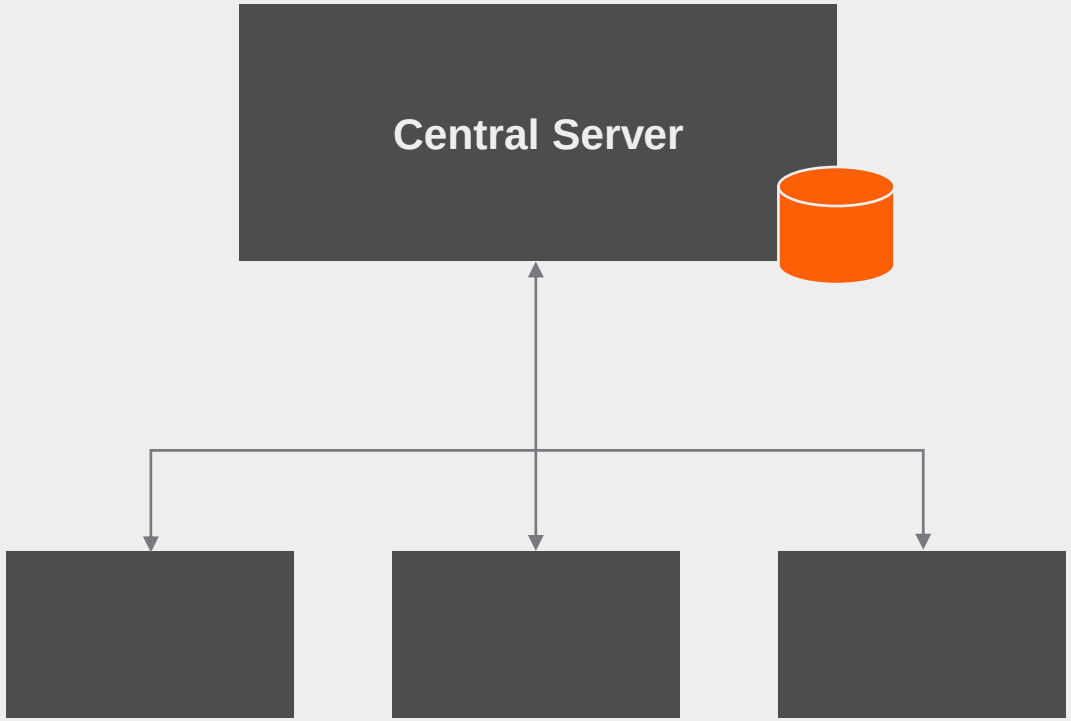
- Created by Linus Torvalds for work on the Linux kernel ~ 2005.
- Some of the companies that use Git:

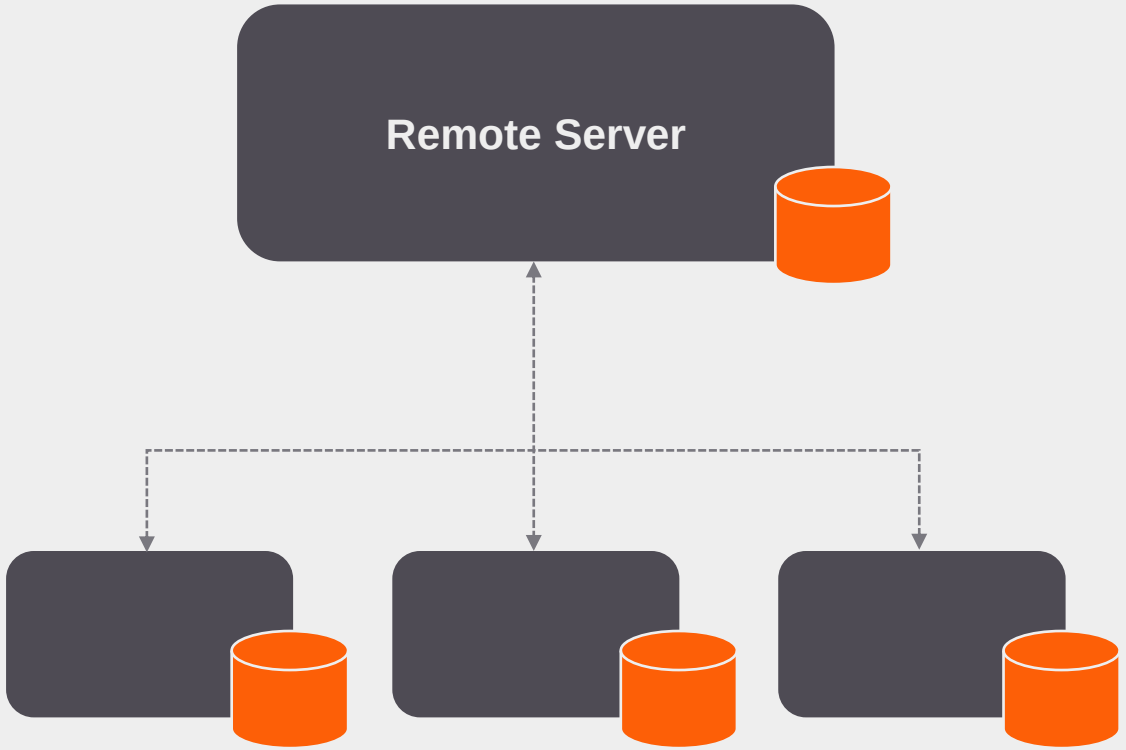
Git is a Distributed Version Control System
What is Git?

Benefits of Git

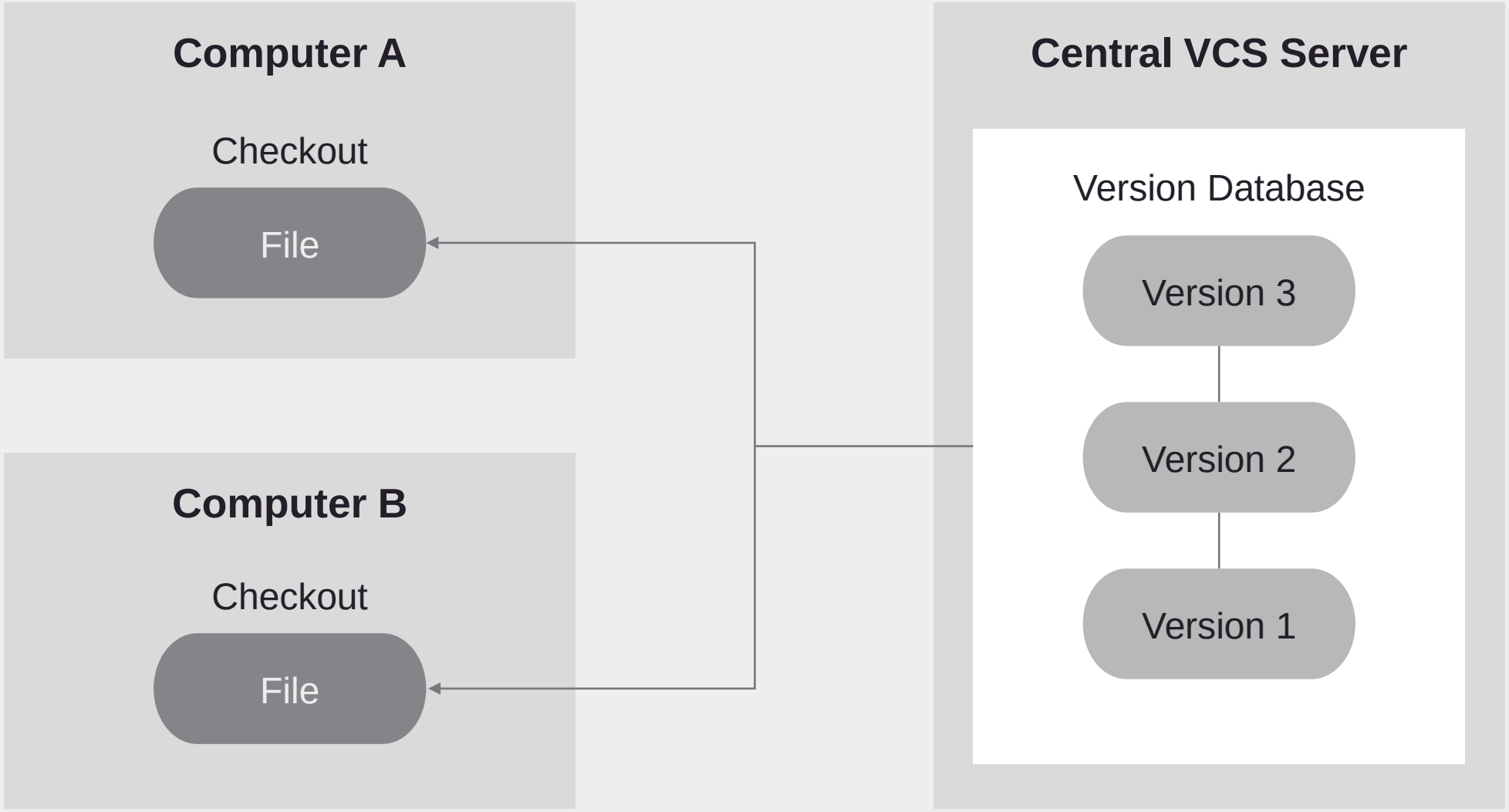
- Git is distributed.
- Most operations in Git only need local files and resources to operate, every operation in Git is local.
- Everything in Git is check-summed before it is stored i.e. It has integrity.
- The Three States – The Git working directory, The staging area and Repository.
- Everyone has the complete history.
- Everything is done offline.
- No central authority.
- Changes can be shared without a server.
- The entire history of the project right there on your local disk, most operations seem almost instantaneous.
- If no access to server or VPN, no need to wait till we get the access because everything is available locally if not take it from your friend (peer).

Centralized VC vs. Distributed VC

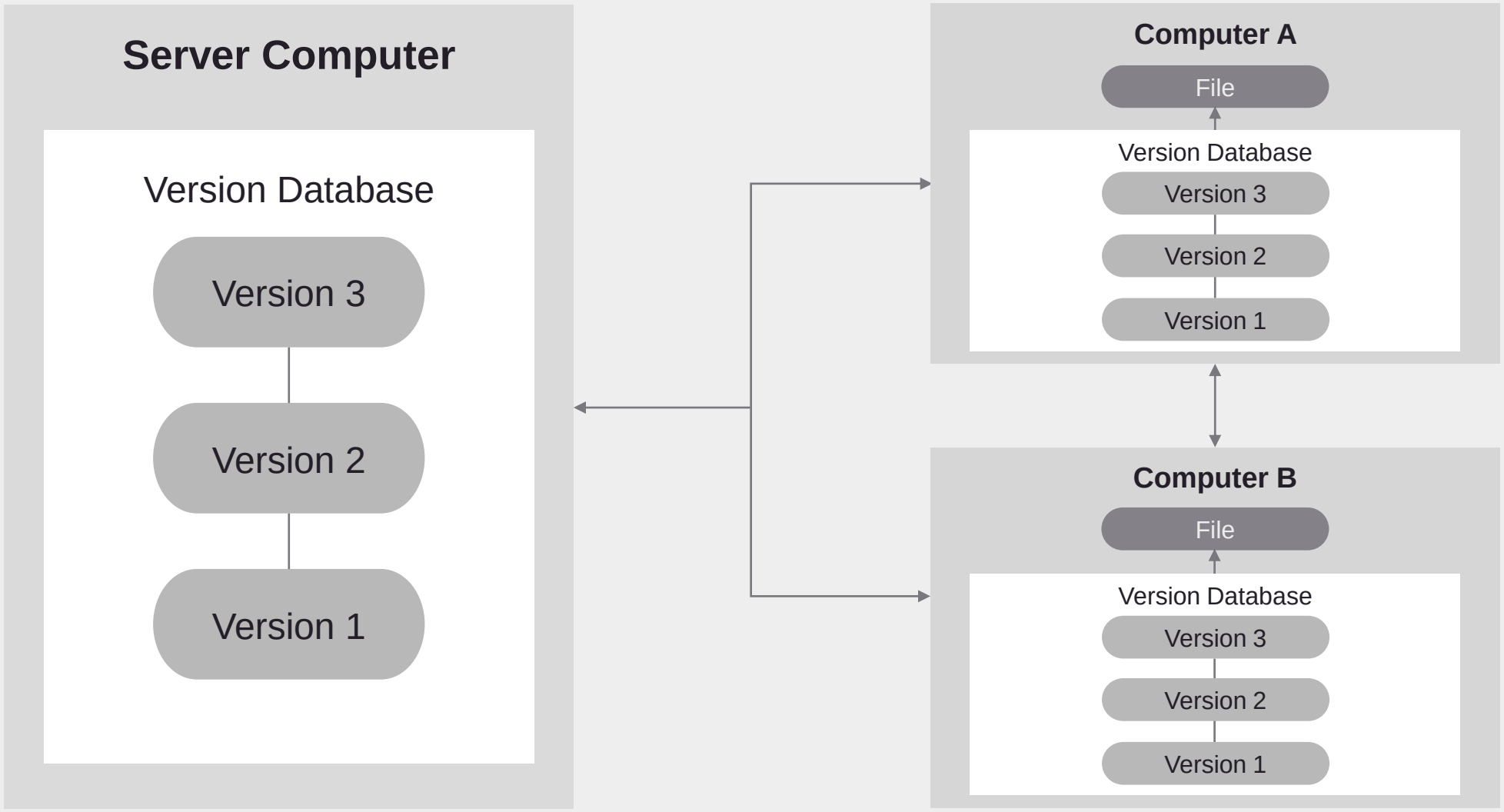




Centralized Version Control

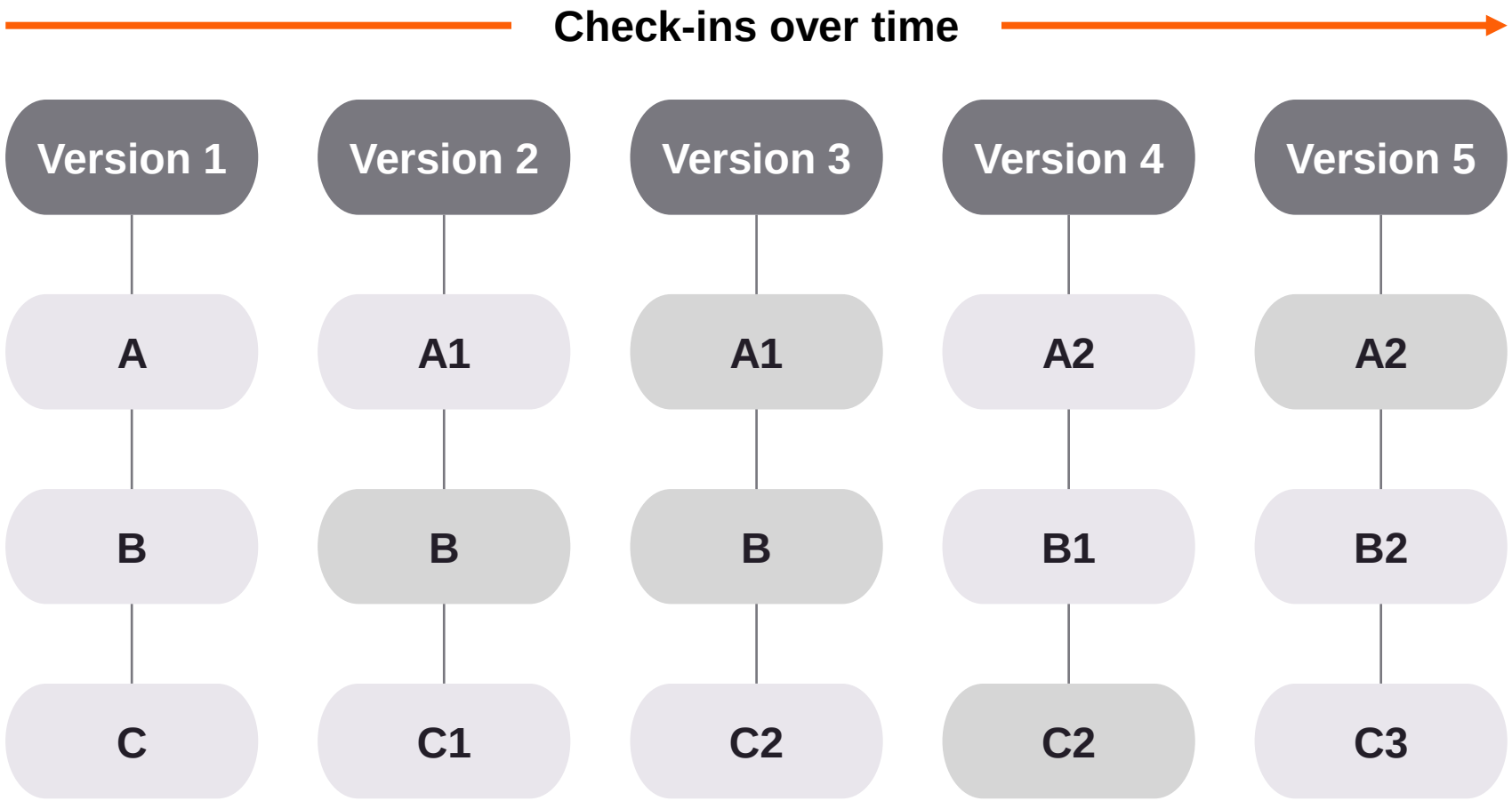


Distributed Version Control

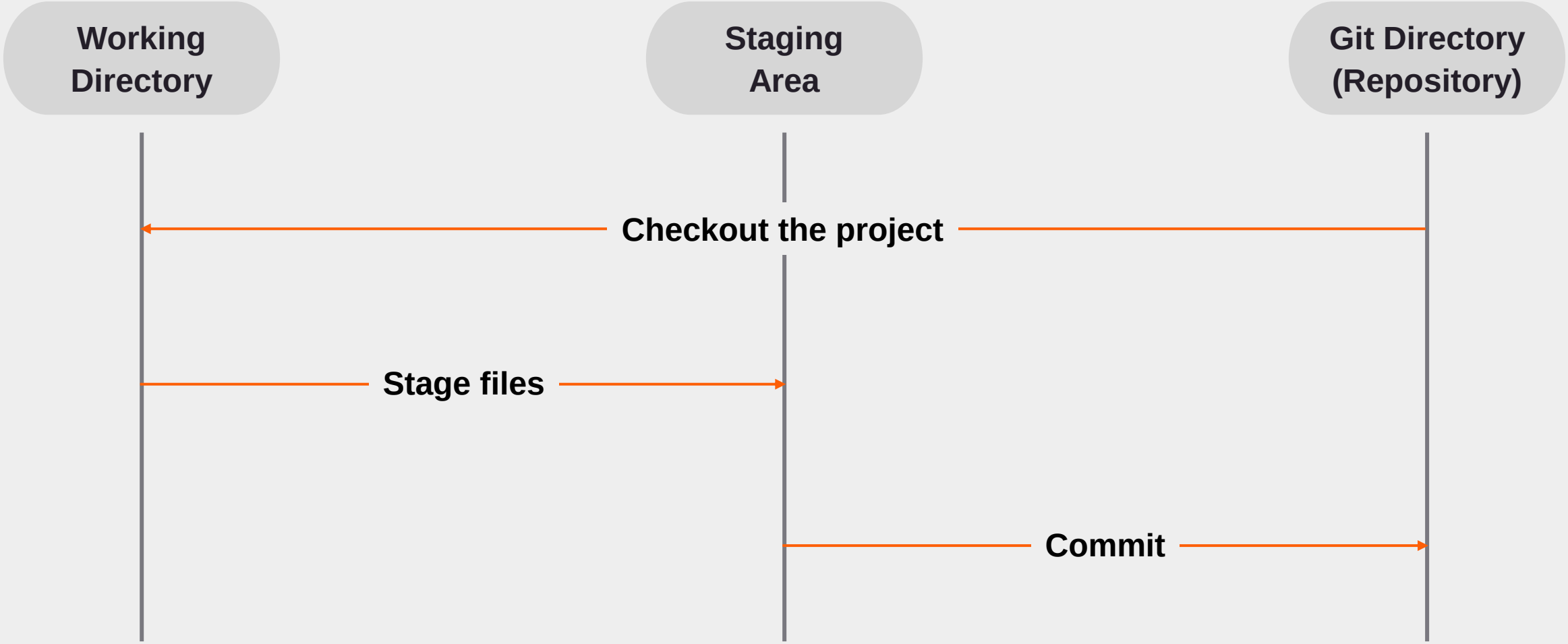


Distributed: Snapshots

- Database is stored on your local machine
- Must “checkout” from database into working directory to edit
- Must “commit” from working directory into database
- Stored in Git database in compressed format
- Files are stored by SHA-1 hash rather than filename
- In this example, files A, B and C are tracked



Why might you want to stage files?



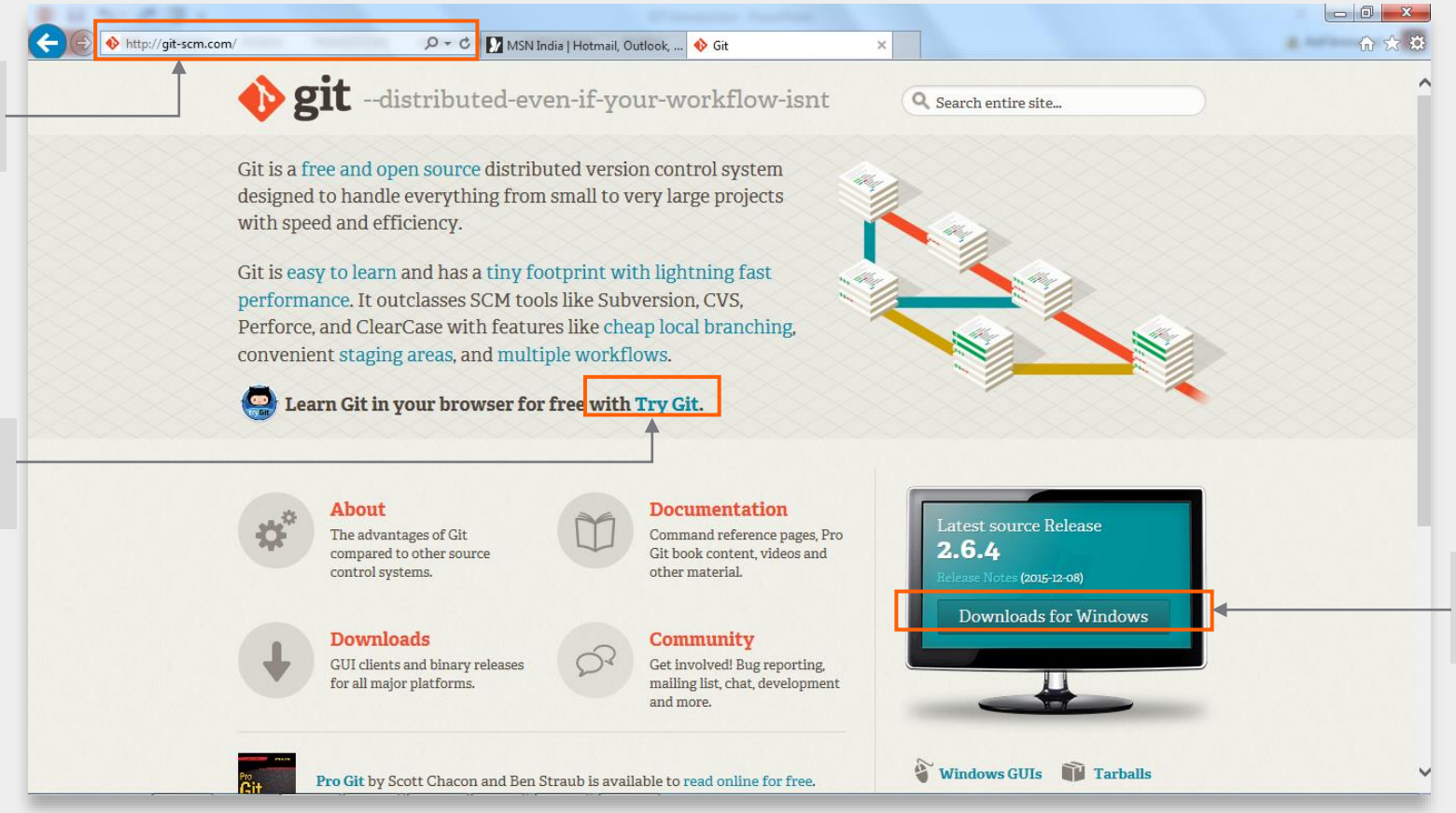
Getting Started: The Command Line

- There are a lot of different ways to use Git.
- There are the original command line tools.
- There are many graphical user interfaces of varying capabilities.
- For now, we will be using Git on the command line.
- Before we start using Git, we have to make it available on computer with below download link:
  - http://git-scm.com/download/win

Installing GIT

http://git-scm.com/

Try Git Online..!!!
Awesome ..!!!



Download Git

FAQs

- What is Version Controlling?
- What is SVN and its Pro’s and Con’s?
- History of Git?
- What is Git?
- Benefits of Git over other Version Controlling Tools?
- How Git works – Three States?

Summary

- With this we have come to an end of our first session, where we discussed about.
  - What is Git?
  - Benefits and working of Git.
- At the end of this session, we see that you are now able to answer following questions:
  - What is Version Controlling?
  - How Git is Distributed Version Control?
  - What are the states in Git?
- In the next session we will discuss about.
  - Working with Git and Local Operations on Git.

Reference Material: Websites & Blogs

- https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control
- https://git-scm.com/video/what-is-git
- https://en.wikipedia.org/wiki/Git_(software)
- http://www.vogella.com/tutorials/Git/article.html
- https://www.siteground.com/tutorials/git/

Pro Git

- By Scott Chacon and Ben Straub
- Publisher: Apress

Version Control with Git

- By Jon Loeliger, Matthew McCullough
- Publisher: O'Reilly Media

Git Interactive

Vaishali Khatal

vaishali_khatal@persistent.com

Asif Immanad

asif_immanad@persistent.co.in

Vice President

Shubhangi Kelkar

shubhangi_kelkar@persistent.co.in

Persistent

Thank you!

Persistent University