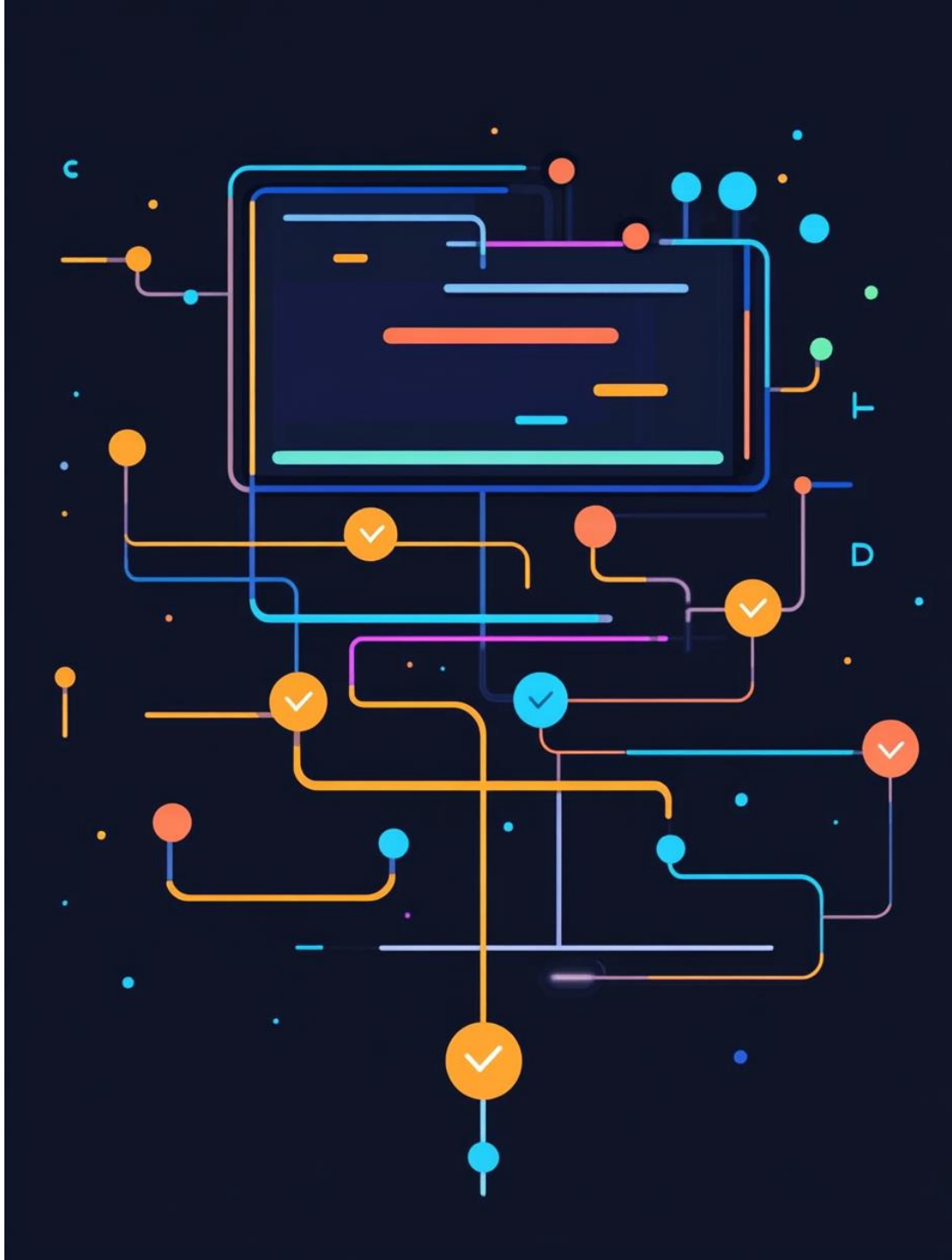
VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Web Application Development

Versioning tools



Ing. Michal Radecký, Ph.D.
www.cs.vsb.cz/radecky

Version control systems

- Version control systems are tools that are used to track and retain changes of code, documents, or any files during the development of a project.
- These systems allow developers and other team members to collaborate on the "same" data.

Benefits of version control systems

- Change Tracking and History Recording
- Restoration based on history
- Collaboration during the development
- Parallel development
- Reduce the risk of errors and conflicts

Types of versioning systems

Local

- They only work on one computer.
- Each developer maintains their own copy of the project.
- RCS (Revision Control System)

Centralized

- All files and their history are stored on a central server.
- Developers work with the files downloaded and uploaded (share with the server).
- CVS (Concurrent Versions System), Subversion (SVN)

Decentralised

- Each developer has a complete copy of the repository, including the history.
- It allows for parallel development.
- Git

Main Functions

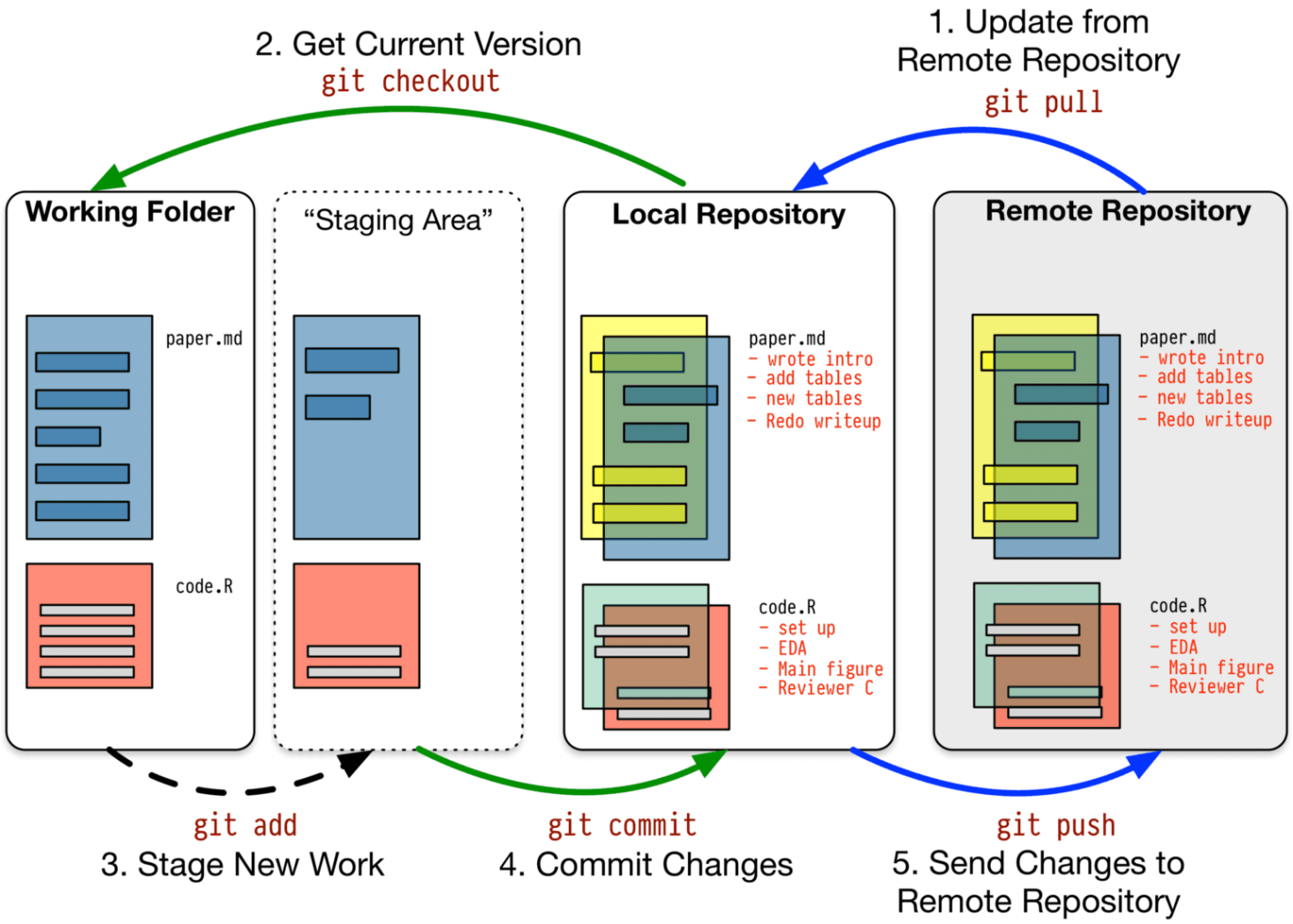
- Commit: Save changes to the repository with a short description of the changes made.
- Branches: Create branches for parallel development without affecting the main branch.
- Merge: Consolidate changes from one branch to another.
- Pull/Push: Update your local repository with changes from the central/remote repository or vice versa.
- Clone: Transfer/clone repositories – create a local one in relation to a remote repository.

Git

- Git is a distributed version control system used for tracking code changes and versioning files during software development.
- It was developed by Linus Torvalds in 2005 and has since become one of the most popular tools of this type.
- Git is open source software and is free to use.

https://chiamakaikeanyi.dev/working-with-git/

Git



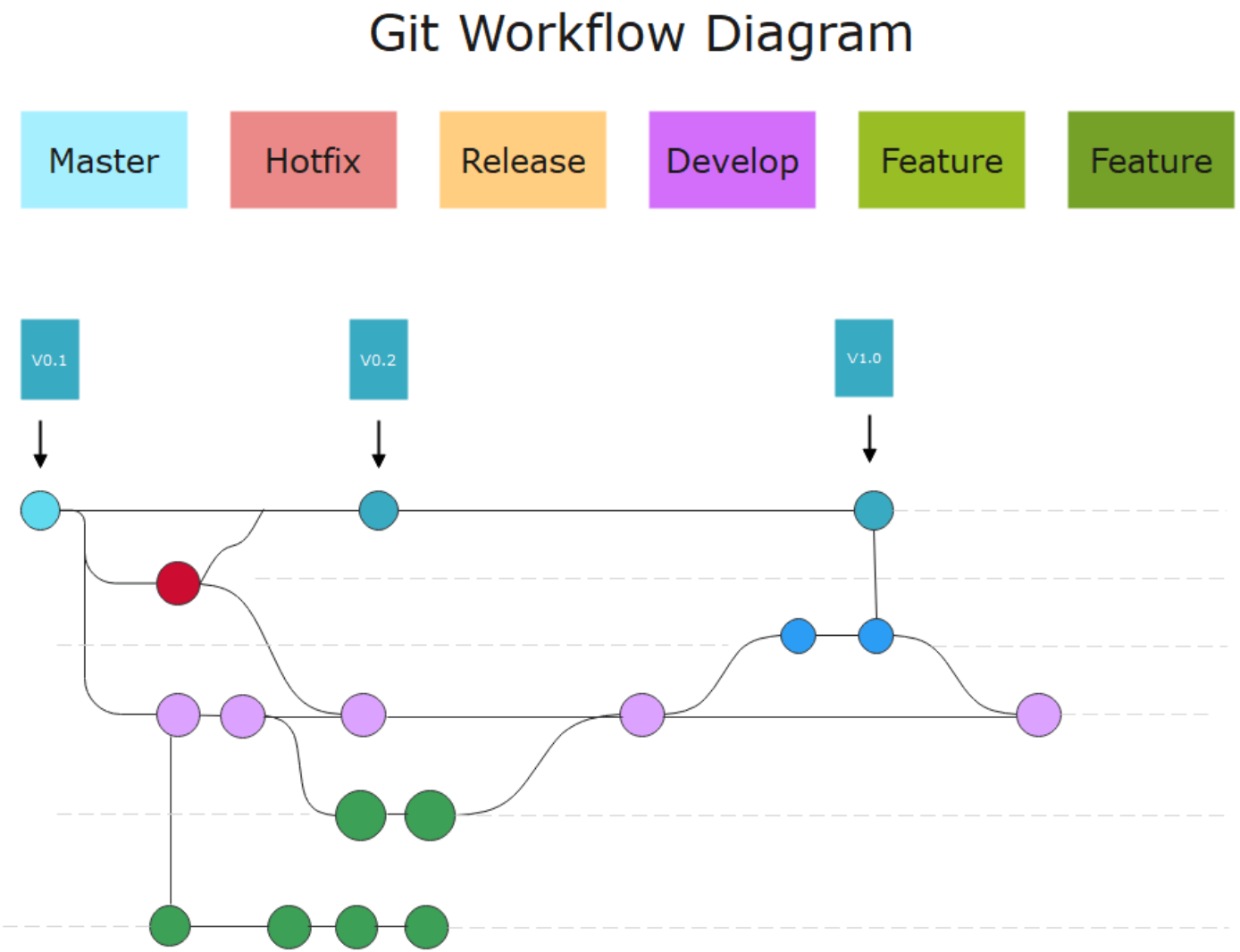
Git

Staging Area

- Index/State Machine - A key concept that allows for precise control over which changes are included in the next commit.
- It acts as a staging ground for the commit.
- Changes that are added here are marked, but have not yet been committed to the local repository itself.
- Tagging can be controlled by "add", which provides flexibility between the draft and the commit.

https://www.edrawmax.com/article/gitflow-diagram.html

Git



GitHub

- GitHub is a web-based platform for version control and hosting of software projects.
- It is one of the most popular repositories (Remote Repository) for developers who want to share their code, collaborate on projects, and track change history.

https://madnight.github.io/githut/#/pull_requests/2023/3

GitHub

Repository
- Users can create repositories for their projects. The repository contains all project files and change history.

Stars and Forks
- Users can rate or create their own projects/repositories from existing ones.

Issues and Pull Requests
- GitHub provides tools for tracking issues and pull requests. This allows for effective communication and collaboration within the team.

Collaborators and Teams
- Users can collaborate on projects by adding collaborators or creating teams with access rights.

Wikis and Projects
- Wikis allow you to create documentation and project pages. Projects offers tools for tracking tasks and organizing work.

Gists
- It allows users to share and discuss small pieces of code, text, or images.

GitHub Actions
- Provides tools for automating processes within a repository, such as build, test, and deployment.

GitHub Marketplace
- A platform for finding and integrating tools and applications into GitHub repositories.

Security a Code Scanning
- Tools for code security and repository security.

Copilot
- Integration to leverage AI in project development and management.

"Package" tools

- These are tools that play a key role in the development of large-scale, typically web projects (javascript).
- Their main task is to ensure effective management of the connections of individual "components" and support of development processes.
- It is based on the principles of the node.js ecosystem.

"Package" tools

- Dependency and configuration specifications (package.json)
- Installing dependencies/packages
- Local vs. Local Global context (workspace)
- Maintenance and management of dependencies and versions (state engine)
- Monitoring of versions within the application under development (lock)
- Linking to public and private repositories (Registries)

npm, yarn, …

Workflow tools

- Tools that allow you to "compile" and ensure that the "compilation" process runs within the framework of web application development.
- It is basically a tool for creating bundlers and task runners.
- They can be linked to specific technologies (javascript, CSS, TS) or general.
- Configuration "scripts" are used to define the required output and procedures.
- They can also cover optimization, modular processing, minification, etc.

Webpack, Esbuild, Gulp, Grunt, Parcel, …

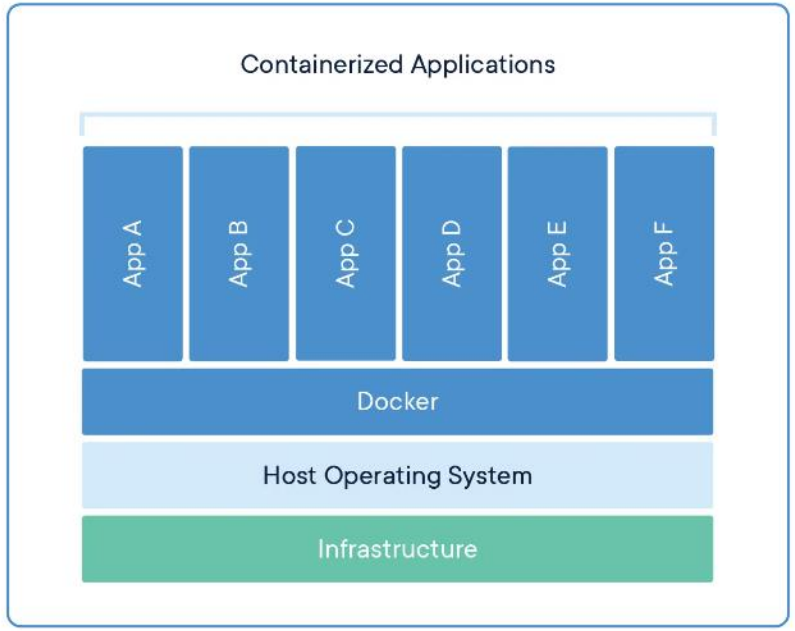
Container runtimes

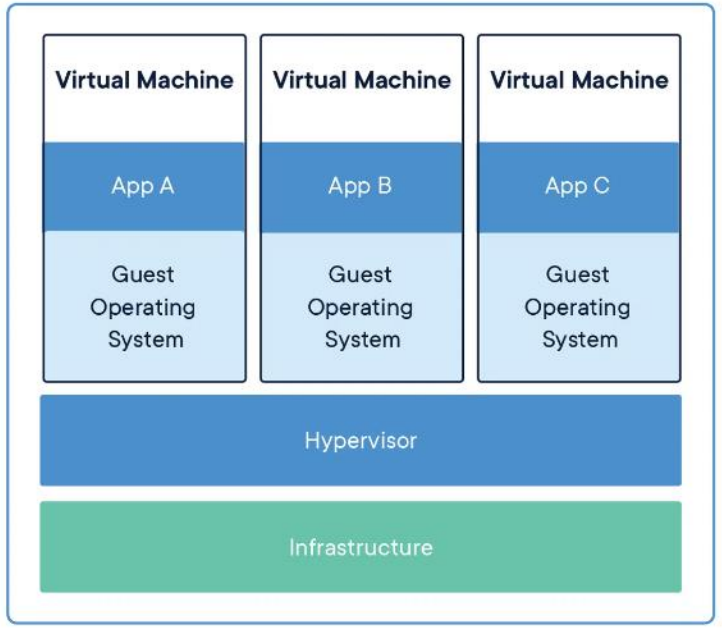
- Platforms for ensuring the ecosystem of a given application runs in a defined environment.
- An isolated container containing a complex application, as well as other tools for its execution (runtime, libraries, dependency data, etc.)
- It is not virtualization, but it can be used including infrastructure.
- They include packaging, distribution, and operation of the runtime environment.

Docker, Kubernetes (orchestralization)

Docker

- Docker has become the standard in containerization and is widely used for deploying, testing, and managing applications in a variety of environments.
- It is popular for its lightness, portability, and ability to quickly and repeatably deploy applications in standardized containers.





Docker

Docker Image

- Docker Image is the basic building block. It is a snapshot of the application, including all the necessary files, settings and dependencies for its run. Images are defined in files called Dockerfiles that describe how to create an image.

Docker Compose

- A tool using a configuration file (YAML) to run a multi-container environment (services, infrastructure, resources).

Docker Container

- Docker Container is a running instance of a Docker Image. A container creates an isolated environment for the application in which it can run independently of the host system. Containers are executable, shareable, and repeatable, ensuring that the application runs consistently across different environments.

Docker Registry

- Docker Registry is a repository for sharing Docker Images. They can be internal or public (Docker Hub).

Dockerfile

- A Dockerfile is a file containing instructions for creating a Docker Image. Describes how to install and configure the necessary components to run an application in a container.

https://www.youtube.com/watch?v=_dfLOzuIg2o