



COMMUNITY SUMMIT

AX/FINANCE & OPS

San Antonio, Texas
October 13-17, 2024



2024

**COMMUNITY
SUMMIT**

Microsoft Dynamics

AX/Finance & Ops

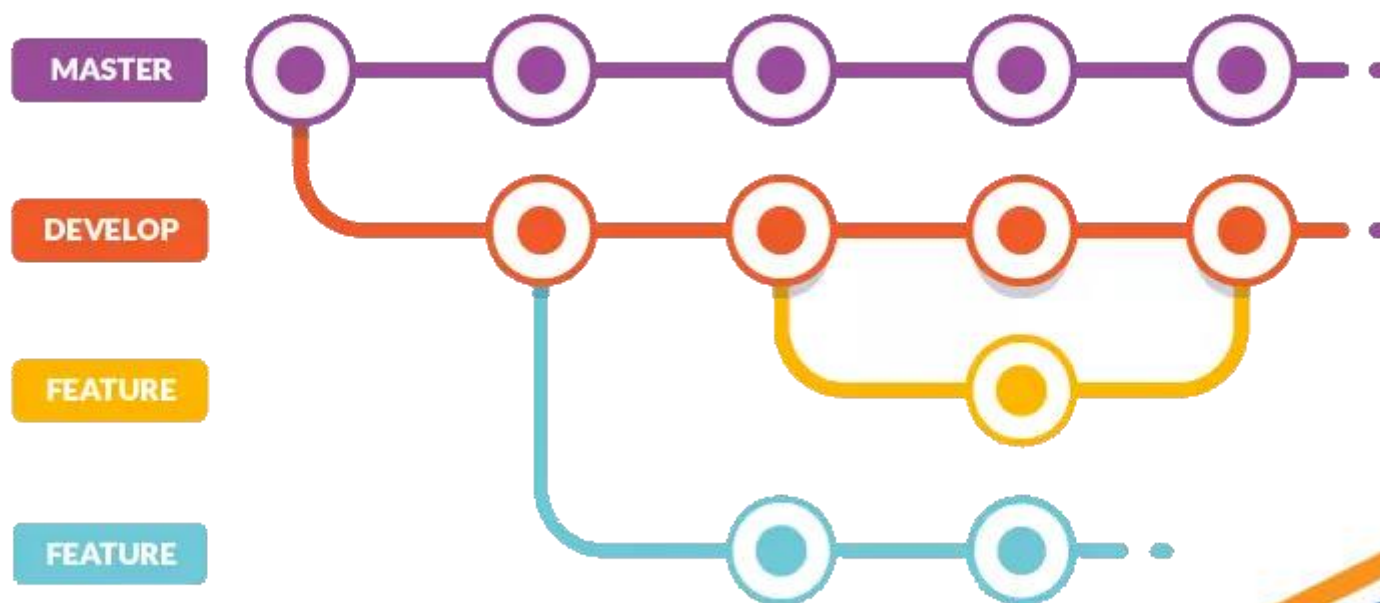
Education
Speaker

The largest independent innovation, education,
and training event for Microsoft Biz Apps



Multiple Projects on D365FO Onebox Using Git

and learn Git along the way...

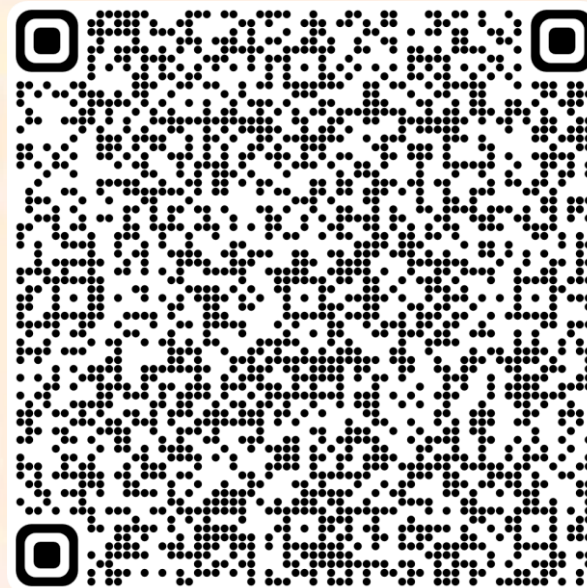


Klemen Novak



CTO @ Docentric

Sometimes I still get my hands dirty and write code ;)



| Agenda

- Intro
- Getting started with Git
- Branching strategies
- Pull Requests
- Merging strategies
- Demo
- Q&A



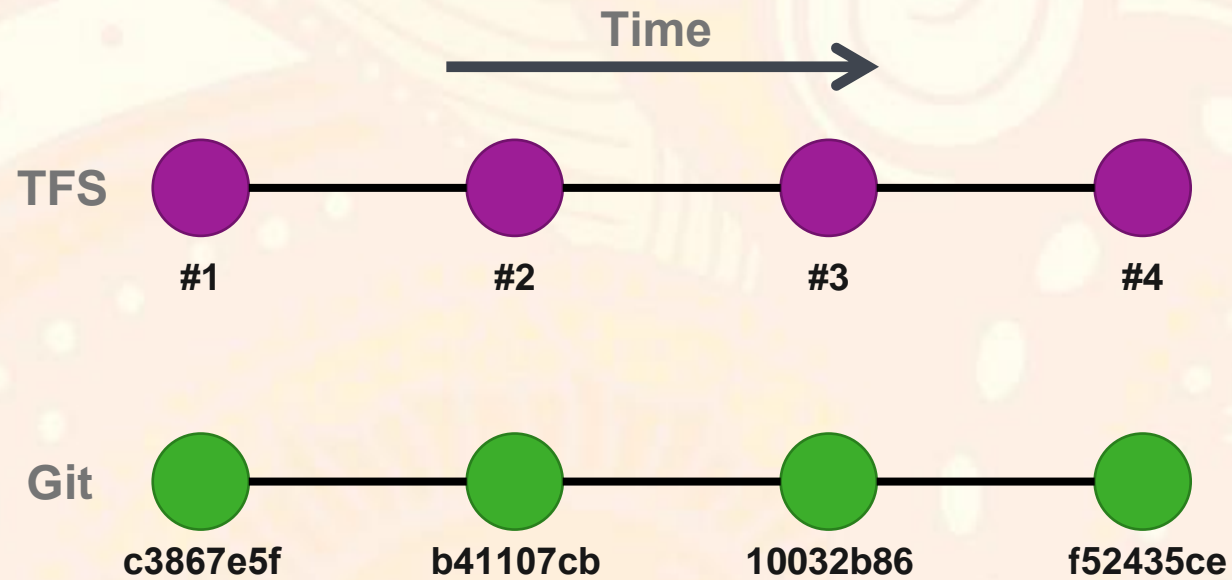


What is Git?

Git repositories:

- **Azure DevOps**
- GitHub
- BitBucket
- GitLab

TFS vs Git Commits



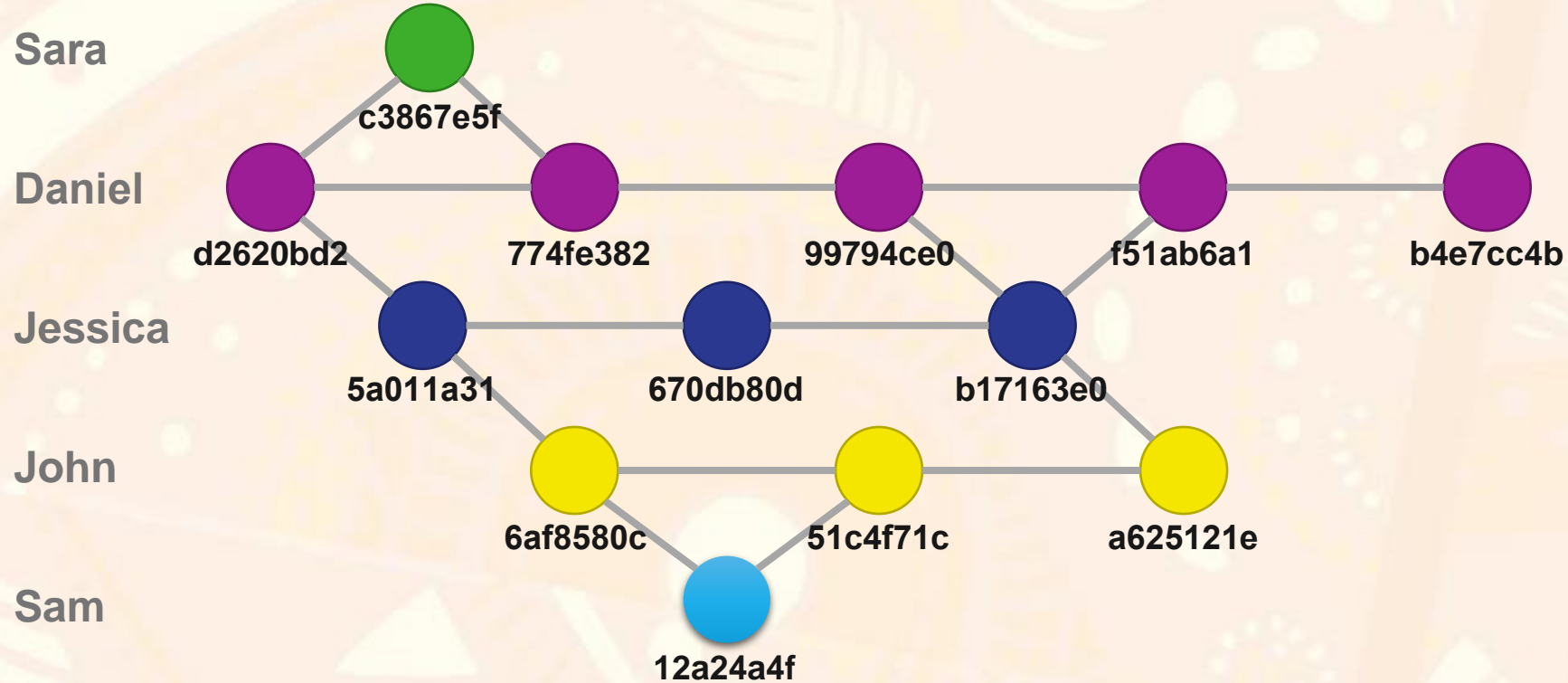
Git commit is a hash:

f52435ce2ffeb7d6b8f1573ca8a6bba9d0697520

Git: Distributed version control

Repository

Time / Feature / Project →



The comparison: TFS vs Git

TFS

- Push based
- Branches last nearly forever and are hard to work with
- Branches are folders based
- Monolithic
- Supports multiple destination folders

Git

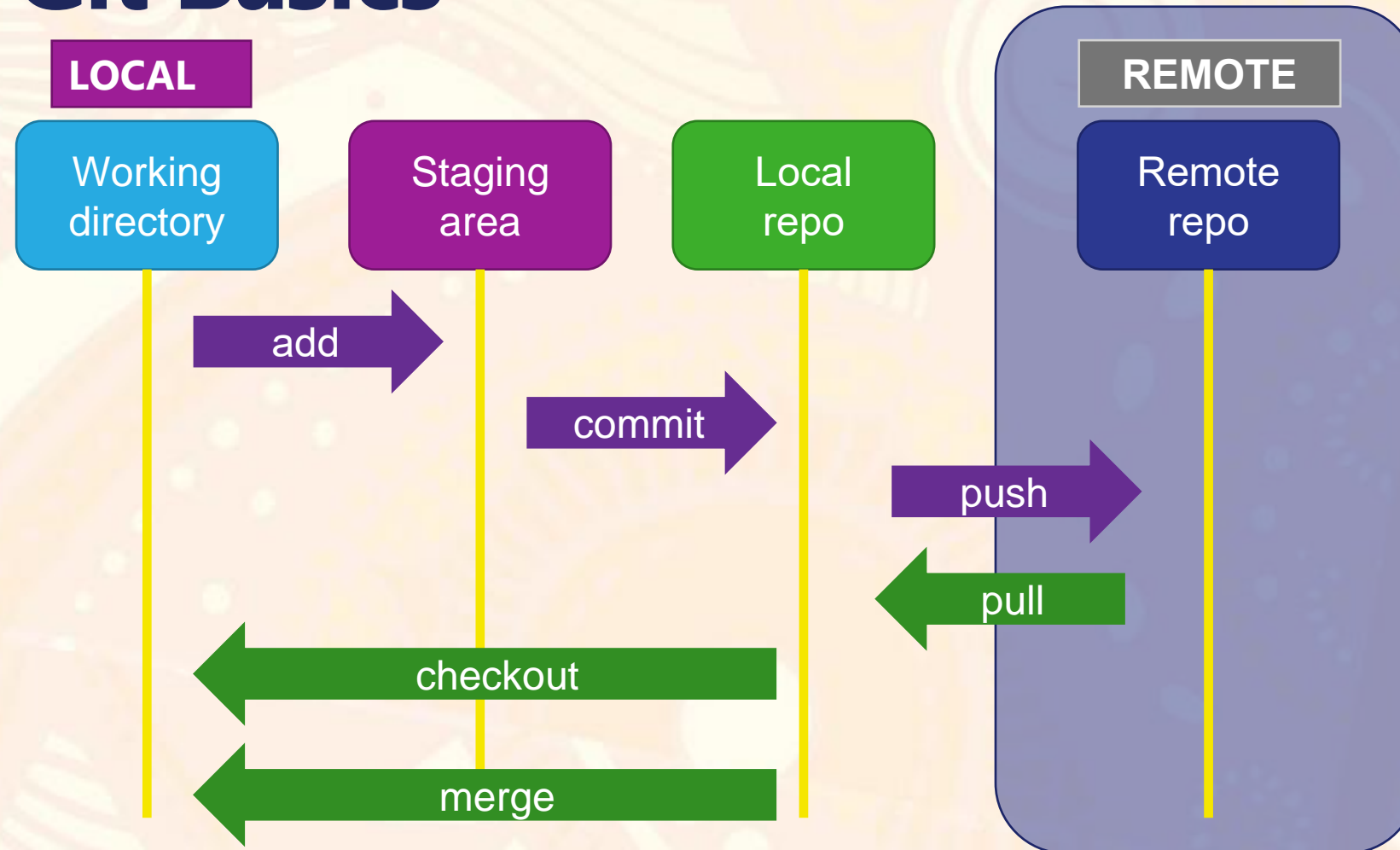
- Modern & Agile
- Working offline (speed)
- Pull-based (Code reviews in one place)
- Branches are short-lived and are easy to work with
- More tools, possible build and deploy automations
- Requires reallocating projects in a single location

Git Basics

A yellow megaphone with a blue handle is shown. Three musical notes are coming out of the megaphone. A thought bubble is connected to the notes, containing the text "Tell me more ;)".

**Tell me
more ;)**

Git Basics



Git Basic Commands

SETUP

Configuring user information used across all local repositories

```
git config --global user.name "Jonh Doe"  
git config --global user.email "email"
```

INITIALIZE GIT REPOSITORY

Initializing local repository (inside a folder)

```
git init
```

Cloning remote repository to local repository (by default, the folder will be named the same as the remote repo)

```
git clone [url]
```

Git Cheat Sheet: <https://education.github.com/git-cheat-sheet-education.pdf>

STAGE & COMMIT

Check the current state of your branch.

```
git status
```

Add all file changes for next commit.

```
git add -A
```

Commit your staged content as a new commit snapshot locally.

```
git commit -m "[descriptive message]"
```


Git Basic Commands

BRANCH & MERGE

List all branches

```
git branch
```

Create a new branch at the current commit

```
git branch [branch name]
```

Switch to another branch and check it out in your working directory

```
git checkout [branch name]
```

Merge the specified branch's history into the current one

```
git merge [branch name]
```

Show all commits in the current branch's history

```
git log
```

SHARE & UPDATE

Add a git URL as an alias.

```
git remote add [alias] [url]
```

Fetch down all the branches from that Git remote (get them listed in the local repository).

```
git fetch [alias]
```

Transmit local branch commits to the remote repository branch (send it to server).

```
git push [alias] [branch]
```

Fetch and merge any commits from the tracking remote branch (get all the changes).

```
git pull
```

Git Cheat Sheet: <https://education.github.com/git-cheat-sheet-education.pdf>

Git Tooling



There are other alternatives.

Getting started with Git



1. Create Git repository
2. Create a new branch
3. Add .gitignore
4. Add .gitattributes
5. Add the code (models and projects)
6. Commit and push to remote

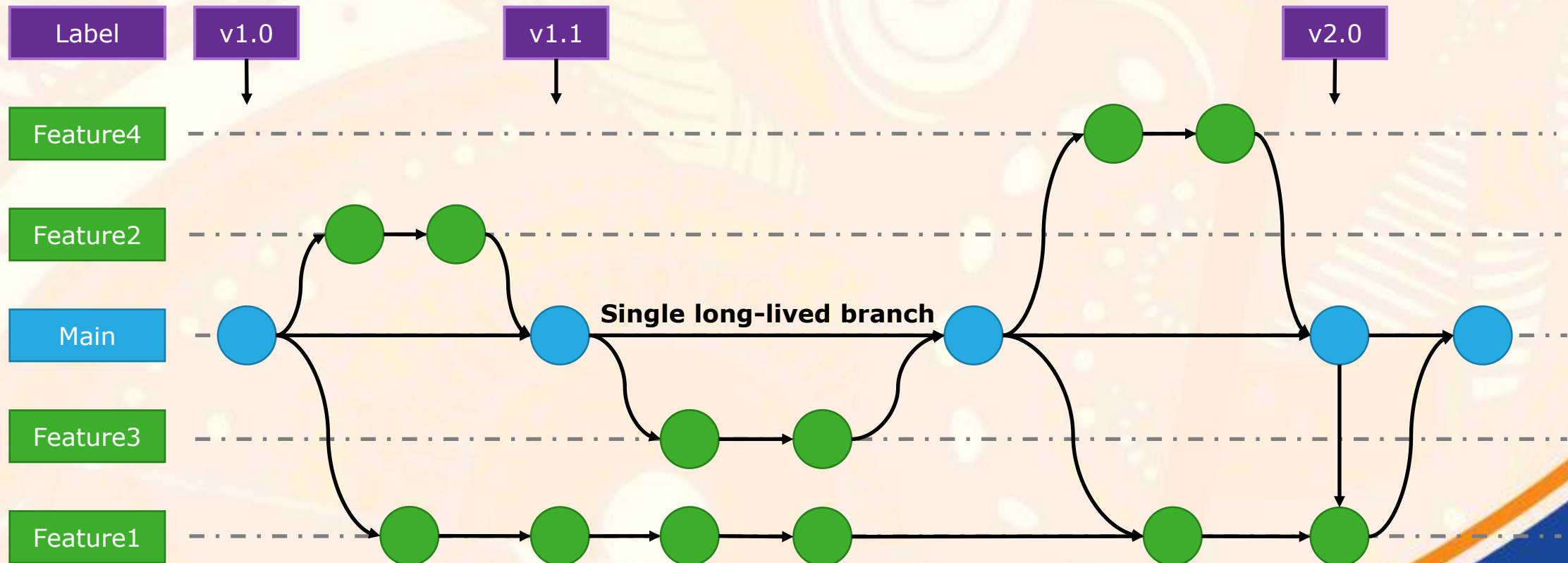
Git Branches



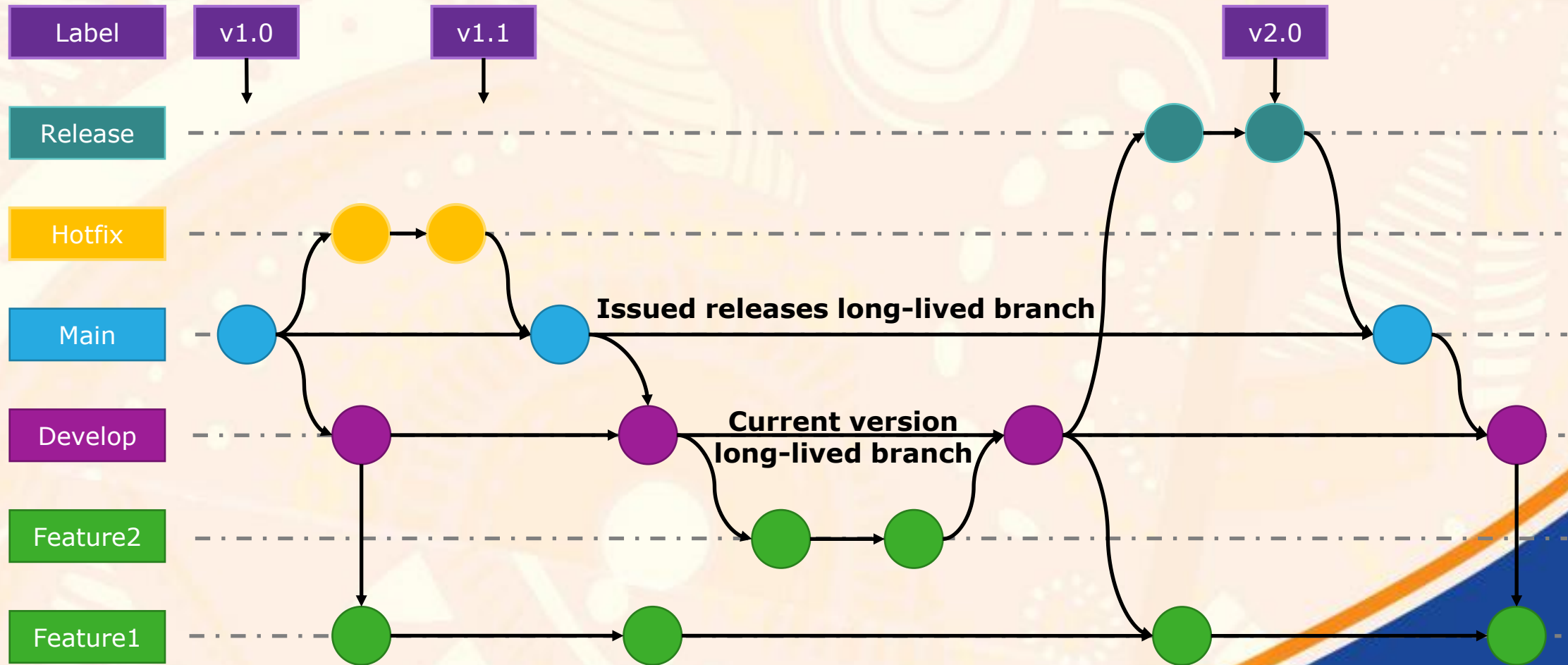
What are
Branches?

How to work
with
Branches?

Branching with GitHub flow



Branching with GitFlow



Branch Workflow Comparison

Github flow

- Simple to use
- A single long-lasting branch (main)
- Intended for each day deployment

GitFlow

- More complex to use
- Two long-lasting branches (main and develop)
- Easier to create hotfixes, new releases
- Intended for product development

Both options are excellent!

The development process is constantly improving and we can change the branch workflow later without hassle.

There are, of course, other branch workflows.

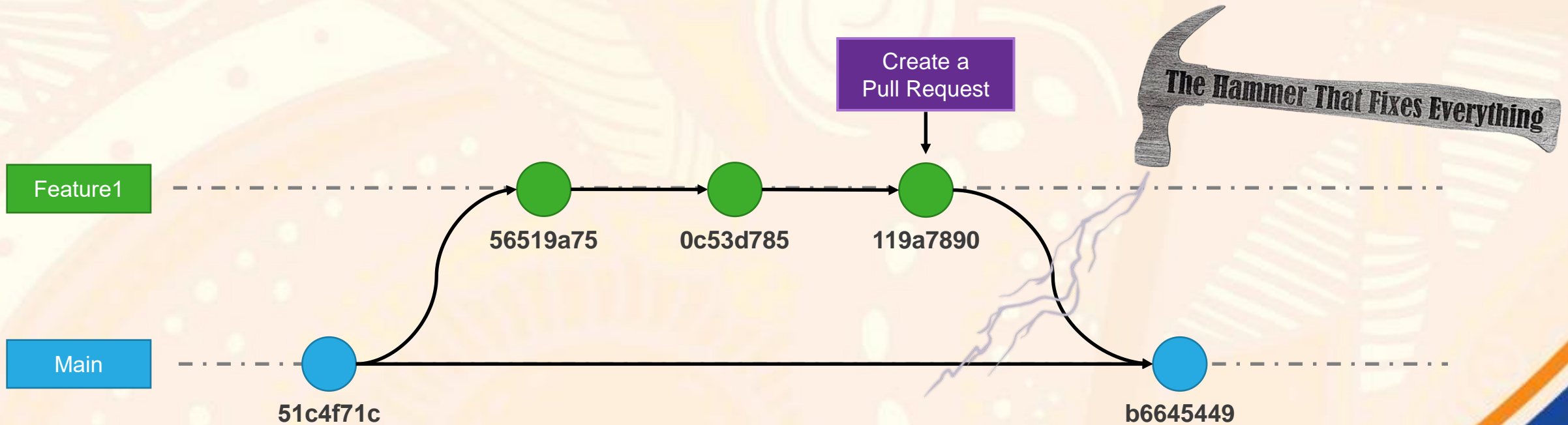
Pull Requests



What are
Pull
Requests?

How are they
related with
branches?

Pull Requests



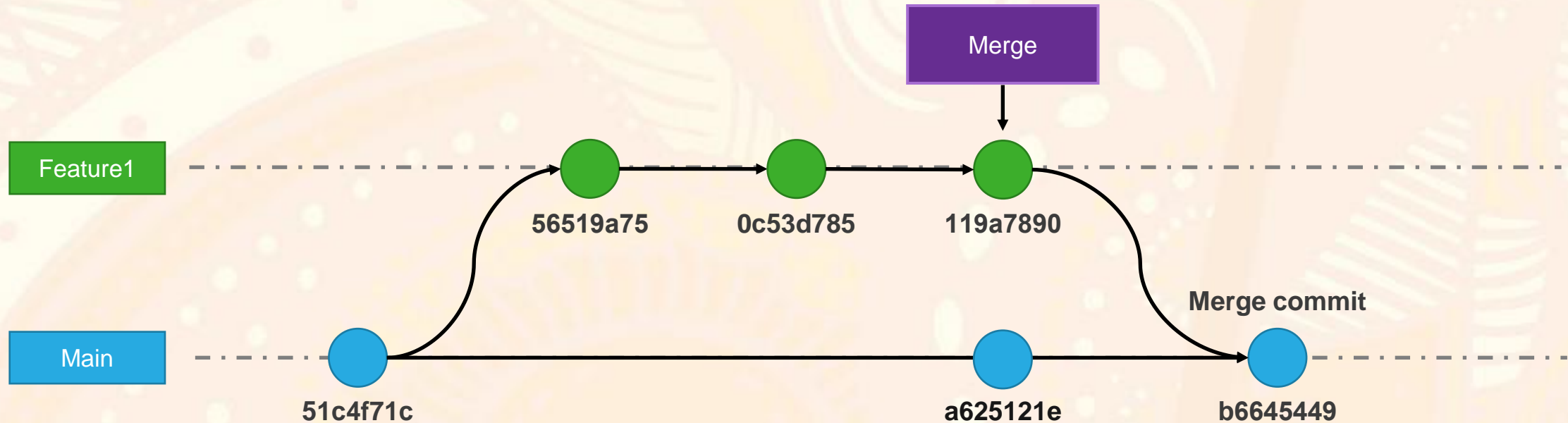
Merging strategies



**What is
Merge?**

How are they
related with
branches?

Merge strategy

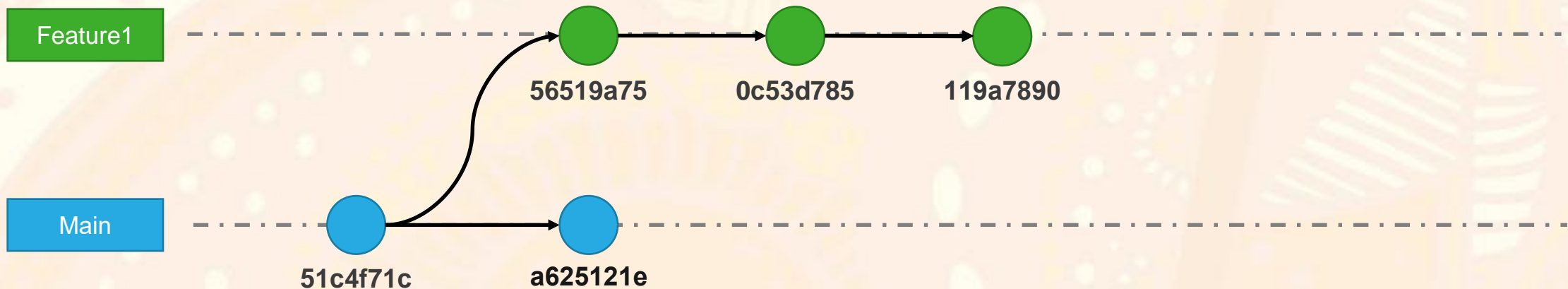


The safe option!

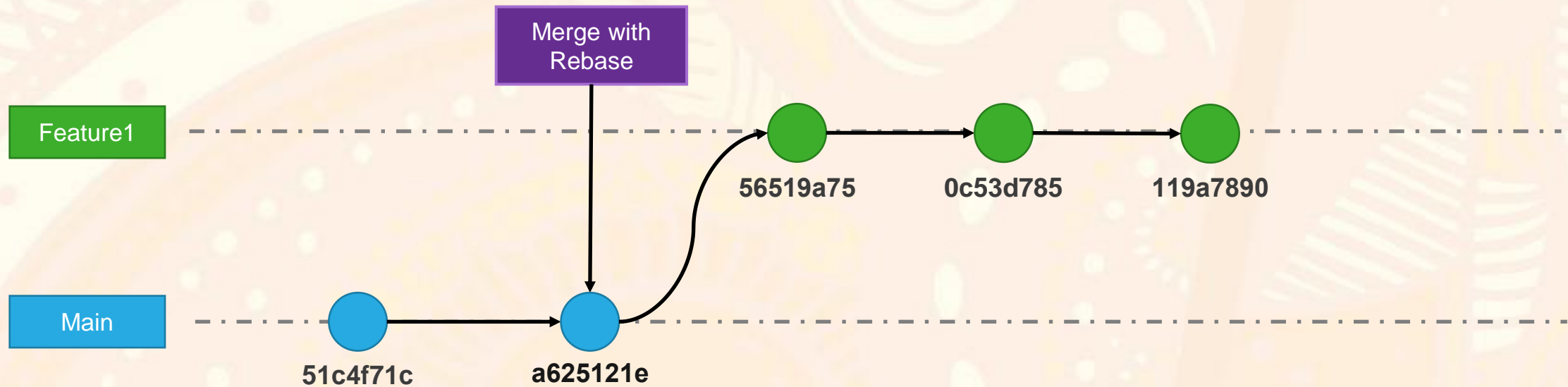
But your history will be messed up by a bunch of merge commits, and your Git graph can end up like an octopus. 🙄

Resolve
conflicts if
needed

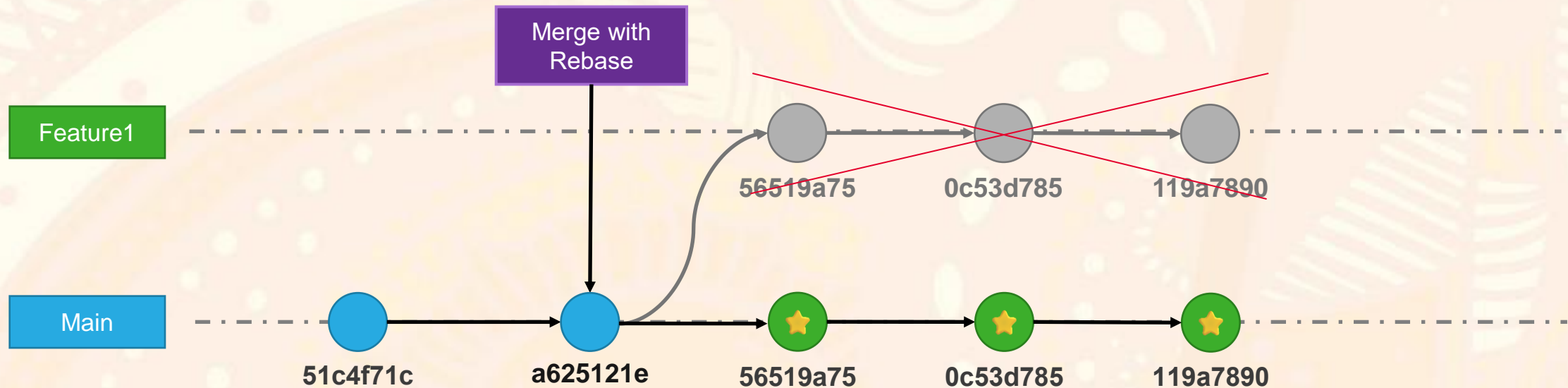
Merge strategy with Rebase



Merge strategy with Rebase



Merge strategy with Rebase

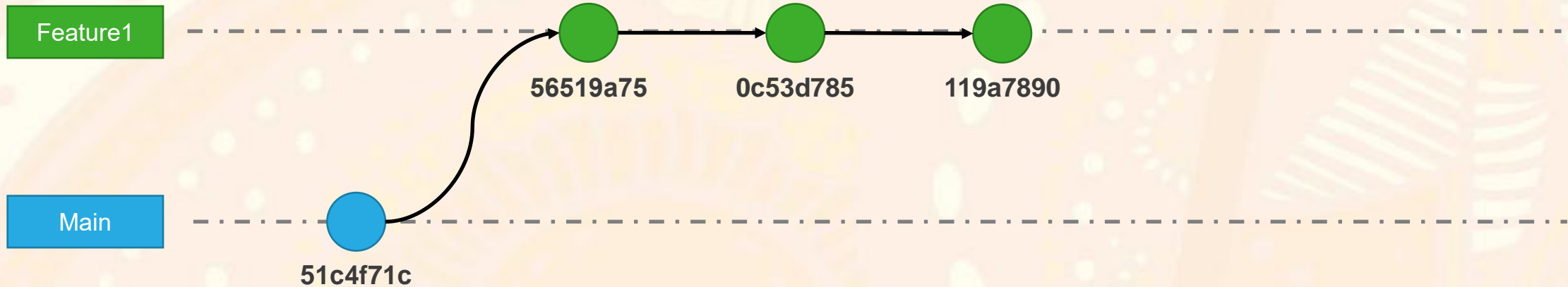


Be careful as you are rewriting history!

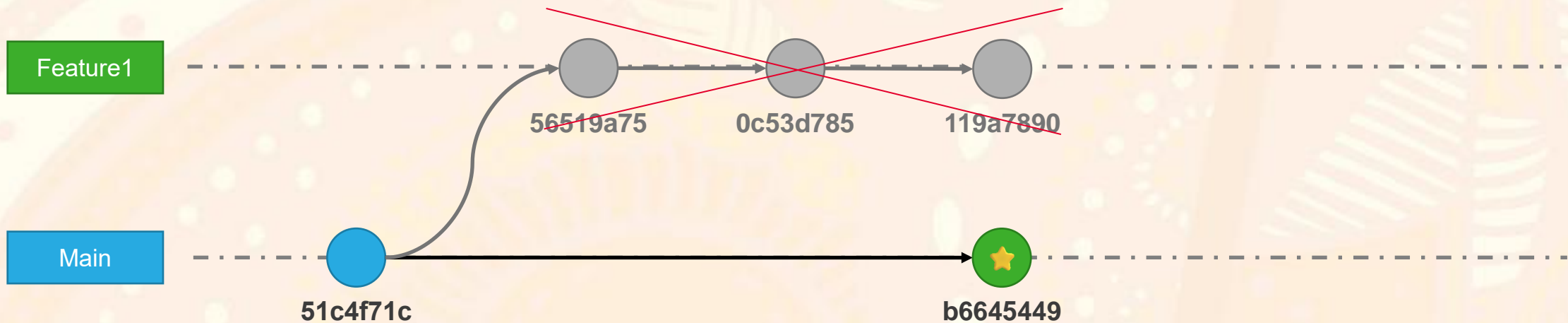
Results in a cleaner, linear commit history that's easier to follow. 👍

Resolve
conflicts
before rebase

Merge strategy with Squash commit



Merge strategy with Squash commit



Great option when doing hotfixes!

Results in a single clean big commit in the main branch, easier to cherry-pick changes to hotfix, but loses detailed history and it's harder to pinpoint the introduced bug. 👍

Resolve
conflicts
before squash



**Finally!
I want a
demo ;)**

The largest independent innovation, education,
and training event for Microsoft Biz Apps

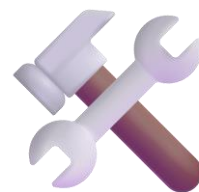


Lessons Learned from D365FO ALM Trenches

Wednesday 16, 8:00 AM, **303B**



Ivan Kaselj
Solution Architect at Docentric



**Visit and
learn more
about CI/CD**

Used resources

- [GitHub Statistics & Trends in 2024](#)
- [Git template repository for Multiple projects on D365FO OneBox and UDE using Git](#)
- [Generate .gitignore files online](#)
- [Gitflow for Visual Studio 2022](#)
- [Nice free Git client: Atlassian SourceTree](#)

Final Thoughts



**That's all
folks!
Thank you!**

Q & A

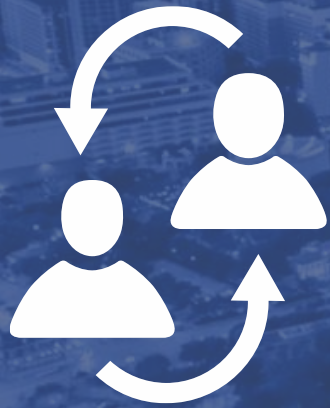


Contact me



OR

**Find me in Hall 1
at Docentric
booth 249**



COMMUNITY SUMMIT

AX/FINANCE & OPS

