



Git Workflow Etiquette

Repository Organization for Clean Codebases

Jolene Langlinais

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source

Agenda

- ↳ About Me
- ↳ Git
- ↳ Branch v. Fork
- ↳ Rebase v. Merge
- ↳ CLI
- ↳ Commit (Messages)
- ↳ Pull Requests (and Reviews)
- ↳ GitHub Issues
- ↳ Open Source



GitHub



Jolene Langlinais

jolenelanglinais.com
@jolanglinais



Maintainer

Full Stack Engineer



ACCORD
PROJECT



CLAUSE



GIT

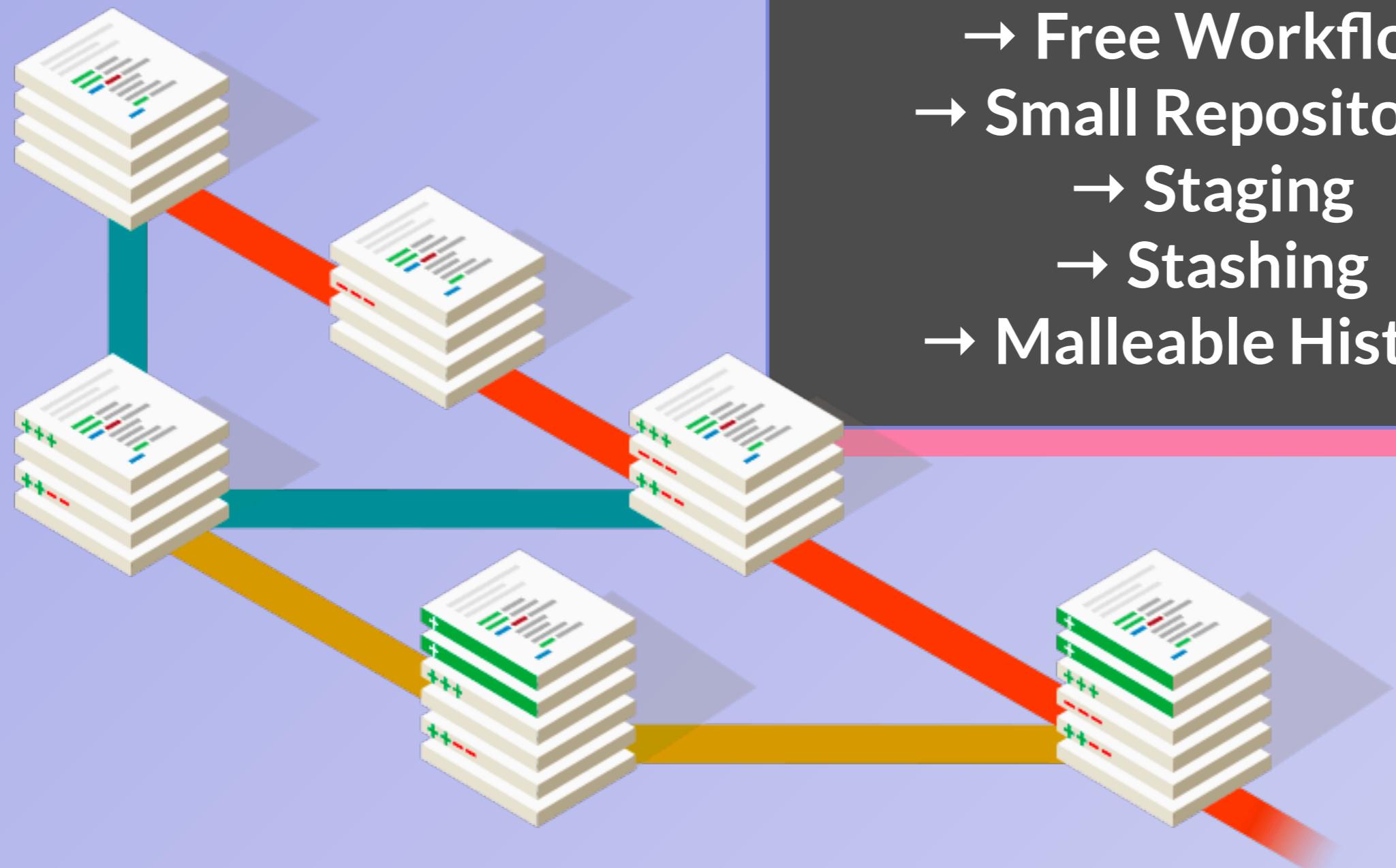
Version Control

Collaboration

Widely Used

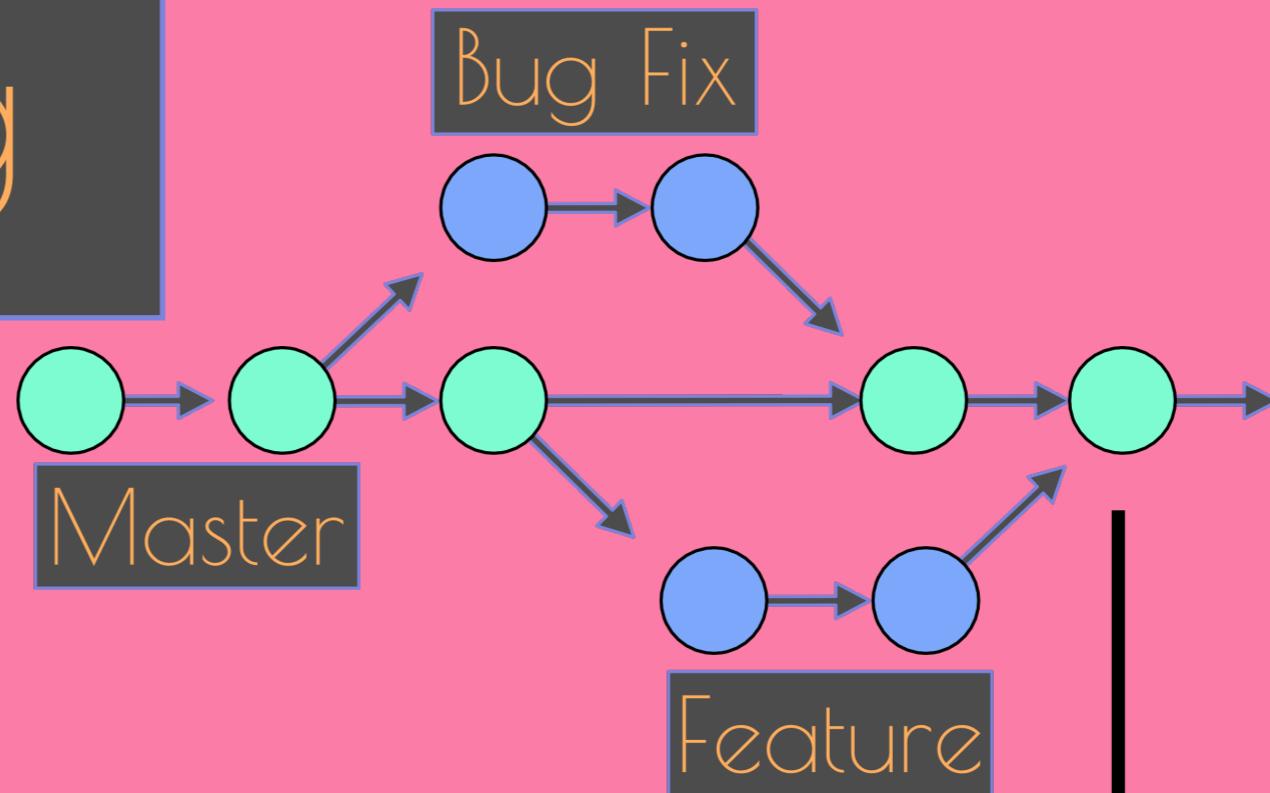
GIT Benefits

- Branching
- Offline
- Content Tracking
- Free Workflow
- Small Repositories
 - Staging
 - Stashing
 - Malleable History

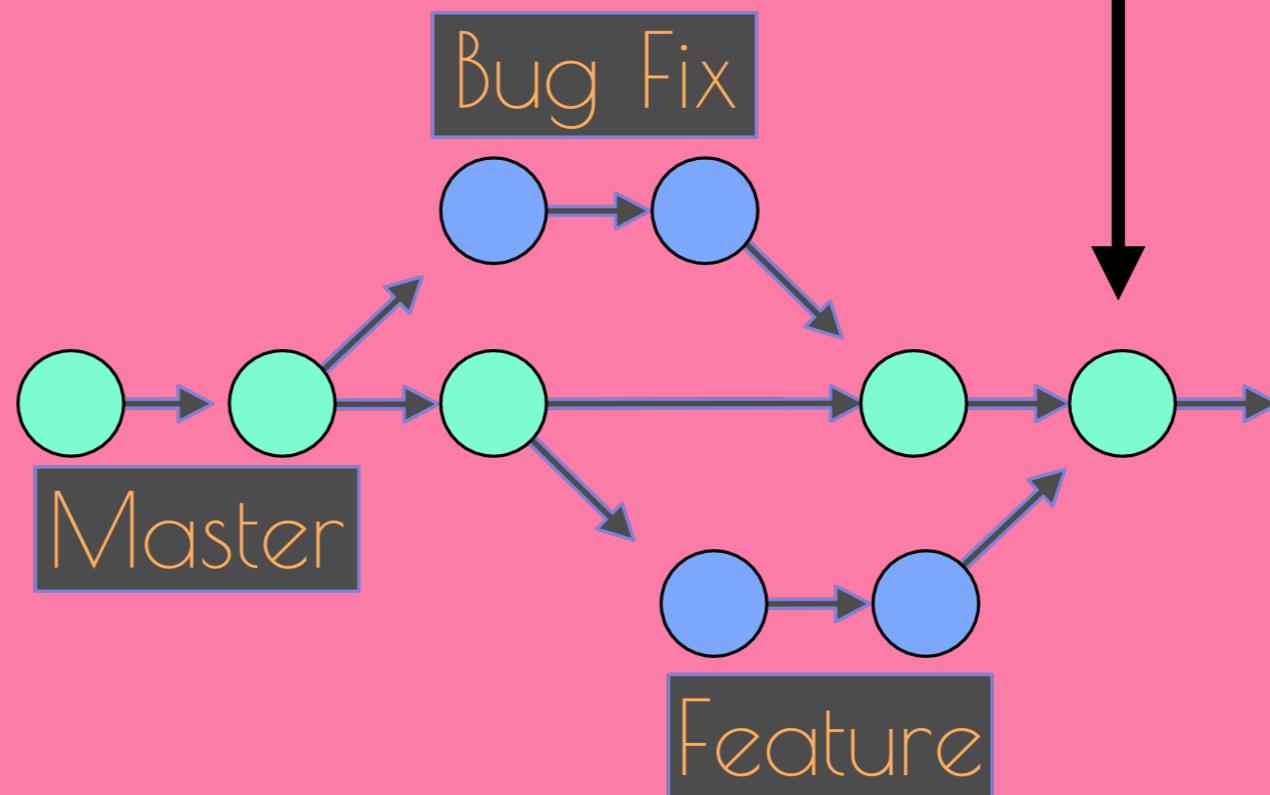


Breaking

FORK



ORIGINAL



Branching

Centralized, trackable branches

Quick and messy

Forking

Incentive to keep clean

Masters need to be synced

Branch Naming

`name/issue-tracker/short-description`

`name`: Anything from initials to full name to GitHub username

`issue-tracker`: Reference the issue from GitHub or some other agile user stories source

`short-description`: One to three words describing the main goal of this branch, separated by hyphens

Branch Naming

```
irmerk/i7/new-feature
```

```
master/i14/routing-service // team branch
```

```
irmerk/i14/routing-service // my branch
```

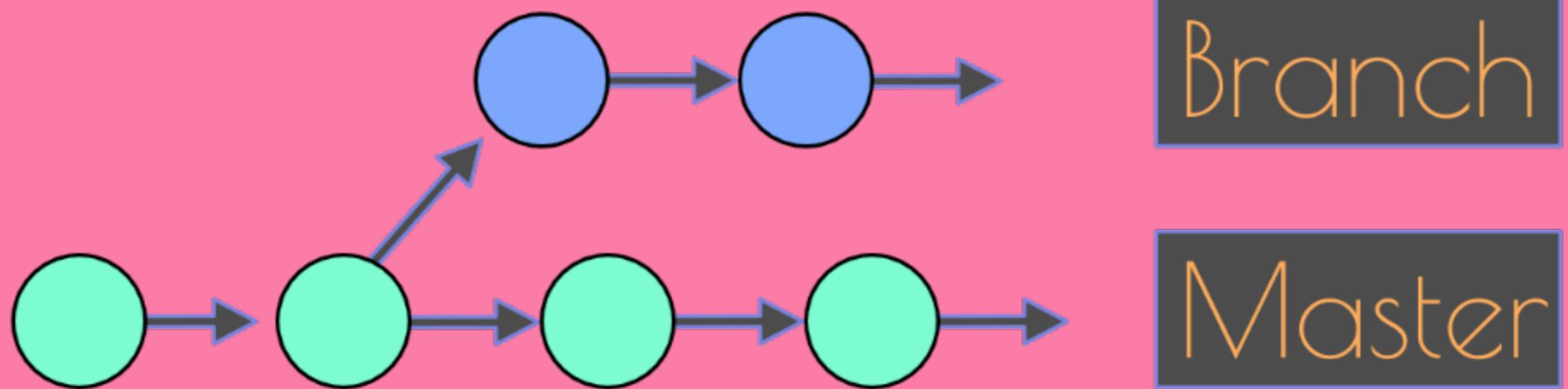
```
someone/i14/routing-service // someone else's branch
```

Rebase

VS

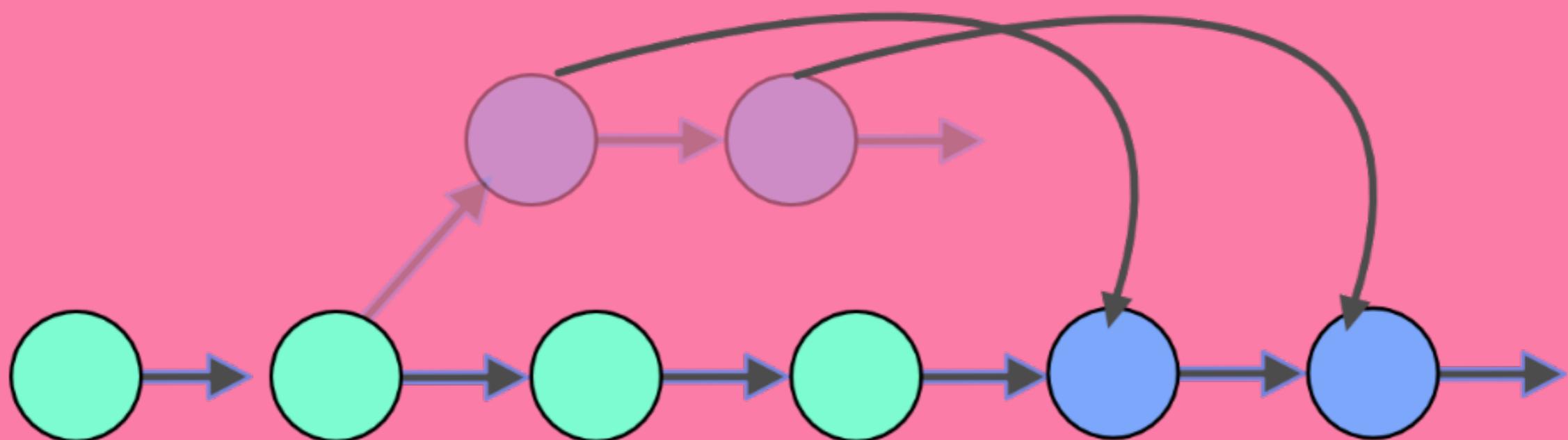
Merge

Rebase v Merge



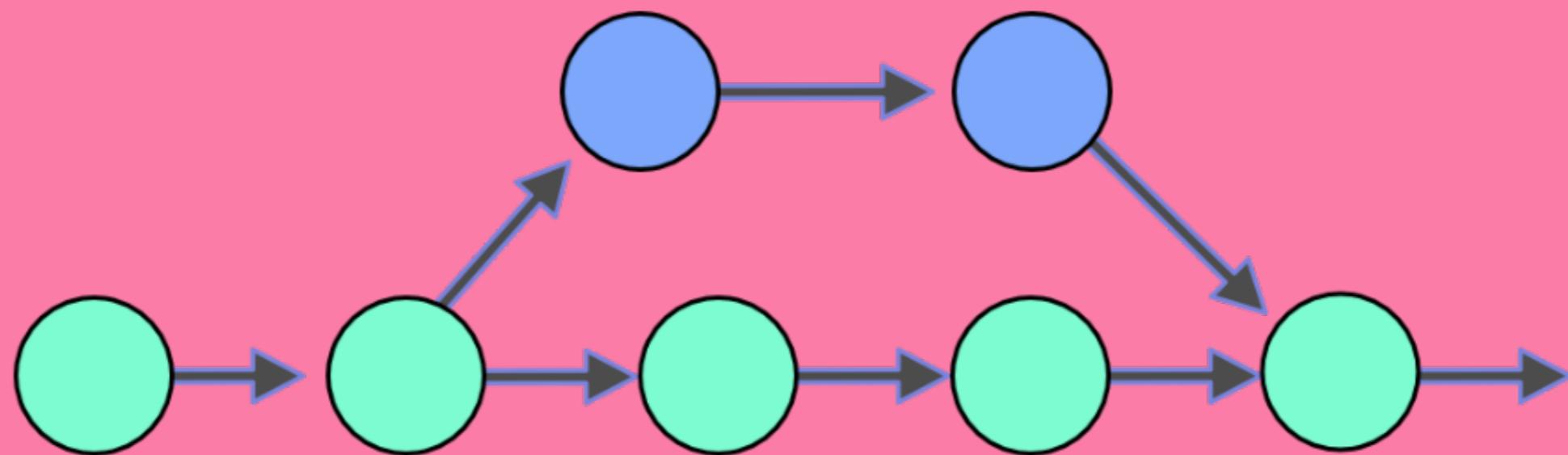
Rebase v Merge

Rebase



Rebase v Merge

Merge



Squash

Reduces clutter

Maintains readable log

Squash

✓ Rebase locally

✓ Squash in GitHub

Squash

🚫 Squash locally

🚫 Rebase after

Interactive Rebase

```
git rebase -i  
  
git rebase -i HEAD~n  
  
git commit --amend -s  
  
git rebase --continue
```

Interactive Rebase

```
git rebase -i  
  
git rebase -i HEAD~n  
  
git commit --amend -s  
  
git rebase --continue
```

Force Push

🚫 CAUTION 🚫

Rewriting History

🚫 CAUTION 🚫

Rebase Your Fears

Commit ongoing work

Practice practice practice

Think of the history!

My Workflow

```
// FORK
git checkout master
git fetch --all --prune
git merge --ff-only upstream/master
git push origin master

// BRANCH
git checkout master
git pull origin master
```

My Workflow

```
git checkout -b branchName  
  
// EXAMPLE  
git checkout -b irmek/i14/react-hooks
```

My Workflow

```
git rebase upstream/master
```

My Workflow

```
git push origin branchName -f
```

// EXAMPLE

```
git push origin irmerk/i14/react-hooks -f
```

My Workflow

```
git branch -D branchName
```

```
// EXAMPLE
```

```
git branch -D irmek/i14/react-hooks
```



Merge



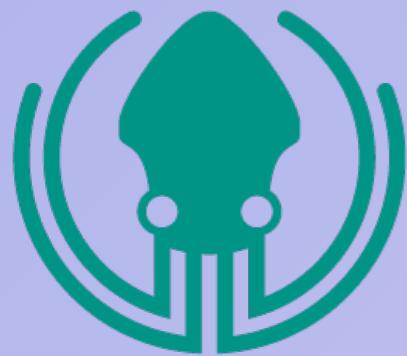
Non-destructive

Easily pollutes log

Command Line Interface



Sourcetree



axosoft

GitKraken



Command Line Interface



Sourcetree



axosoft

GitKraken



Diff

Ensure Code is As Expected

Log

```
git log --graph --decorate --pretty=oneline --abbrev-commit
```

Blame
Specific Lines

Whole Files

Date Ranges

Commit Messages

Commits

Capture Single Logical Changes

Commit Often

Commit Working Order

Messages

Take Your Time

Include Issue References

Use the Terminal

Commit Message Formatting

```
type(scope): subject – footer
```

Commit Message Formatting

Types

`feat`: A new feature

`fix`: A bug fix

`docs`: Changes to only documentation

`style`: Changes to formatting (missing semi colons, etc.)

`refactor`: A code change that neither fixes a bug nor adds a feature

`test`: Adding missing or correcting existing tests

`chore`: Change to build process or auxiliary tools, or maintenance

Commit Message Formatting

Scope

Focal point of new code or best description for where changes can be found.

Commit Message Formatting

Subject

Imperative description of changes, kept under 50 characters (not capitalized and no period)

Commit Message Formatting

Footer

GitHub Issue reference ID

Commit Message Formatting

Examples

```
feat(utilities): include optional state for test utility - I22
  // implement a feature in src/utilities/test/sagaTest.js
```

```
test(sagas): initiate test for failed templates fetch - I22
  // tests run in and of src/sagas/*
```

```
chore(package): install saga testing library - I22
  // edits to dependencies in package.json
```

```
docs(README): add netlify badge and update - I27
  // alter documentation in the readme file
```

Pull Requests

Pull Requests

Share Information

Peer Reviews

Accountability

Pull Requests

Complete Code Addition

Title Format Follows Commits

Squash, Log Persists

PR \leftrightarrow Git

Title Maps to Git

Should be Small

PR \leftrightarrow Issue

Cleanliness

Layers of Single Threaded Goals

Pull Request Formatting

Issue #20

Changes

- Change one
 - Subchange one
 - Subchange two
- Change two
- Theoretically this should be listing all the commit messages included in this PR

Flags

- Possible issues or holds for reviewers to note

Related Issues

- Link any issues or pull requests relating to this

Pull Request

What Changes Address Issue?

What Side Effects?

Why is it Necessary?

PR Reviews
Be Thorough

Guardian of Git History

Keeper of Code Quality

PR Reviews

Obvious Now ≠ Obvious Later

Pristine Code > Finishing Feature

... But Don't Block

PR Reviews

Changes Necessary?

Contributors ≠ Reviewers

GitHub

GitHub Issues

Easily Maintained

Use Liberally

Foundations for Conversations

GitHub Noise

Is Your Presentation Worth It?

Avoid Rushed Answers/Responses

Provide Context!

Open Source Software

Open Source Software



ACCORD PROJECT



Open source software tools for smart legal contracts

accordproject.org

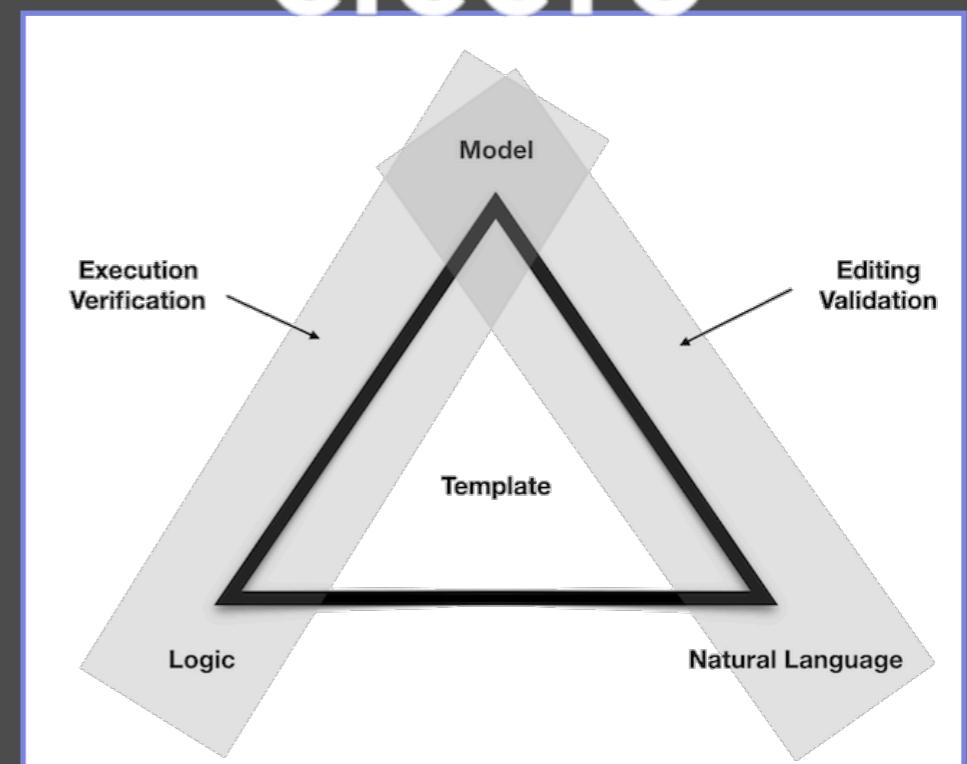
@AccordHQ

CICERO

Natural Language

Model:

Logic:





ACCORD PROJECT



Open source software tools for smart legal contracts

accordproject.org

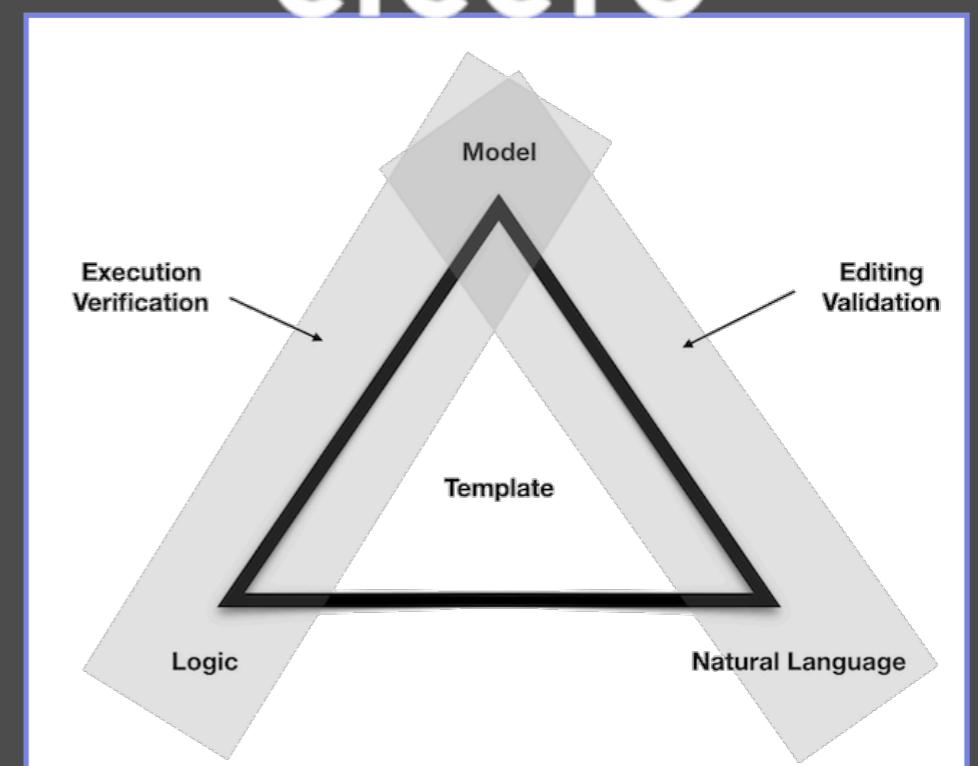
@AccordHQ

CICERO

Natural Language

Model:

Logic:





ACCORD PROJECT



Open source software tools for smart legal contracts

accordproject.org

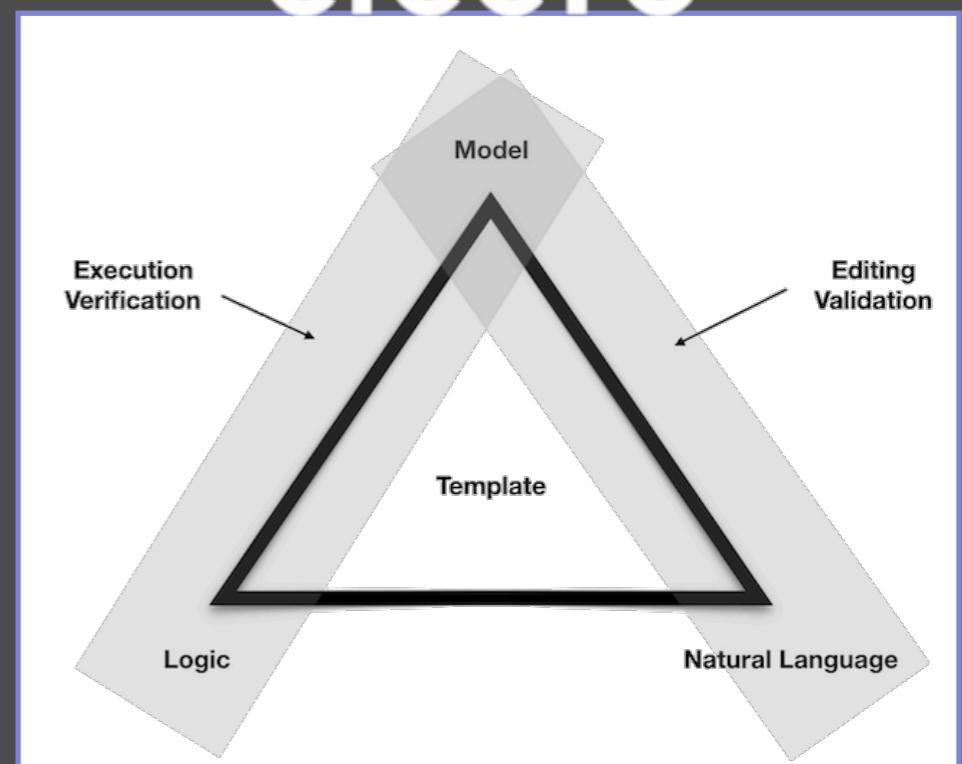
@AccordHQ

CICERO

Natural Language

Model: Composer Concerto

Logic:





ACCORD PROJECT



Open source software tools for smart legal contracts

accordproject.org

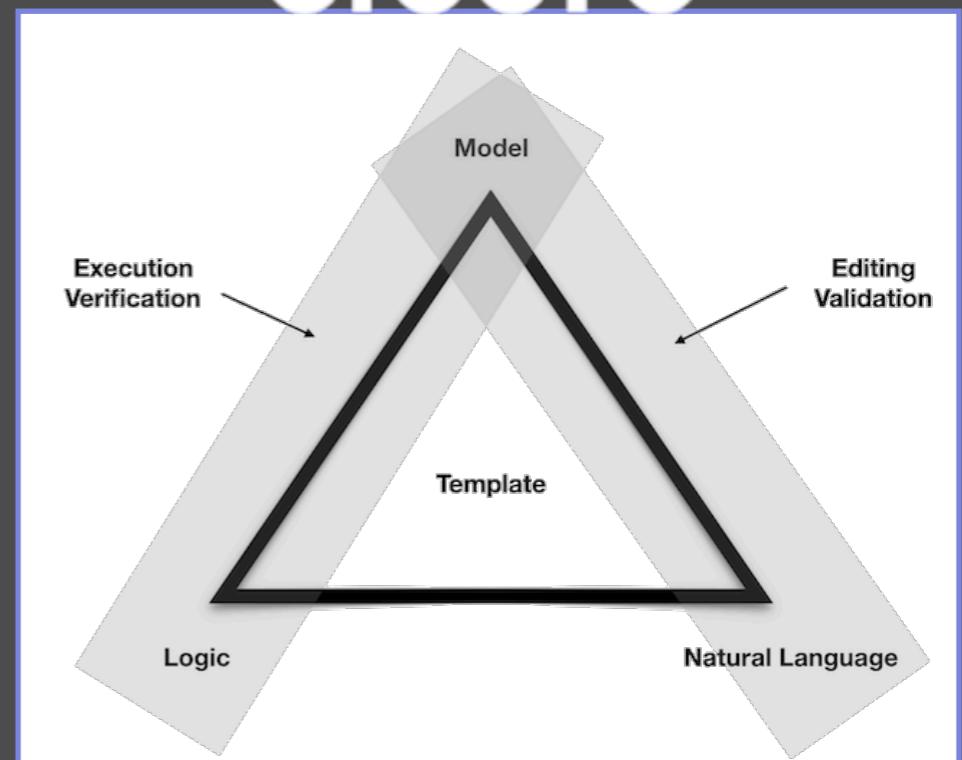
@AccordHQ

CICERO

Natural Language

Model: Composer Concerto

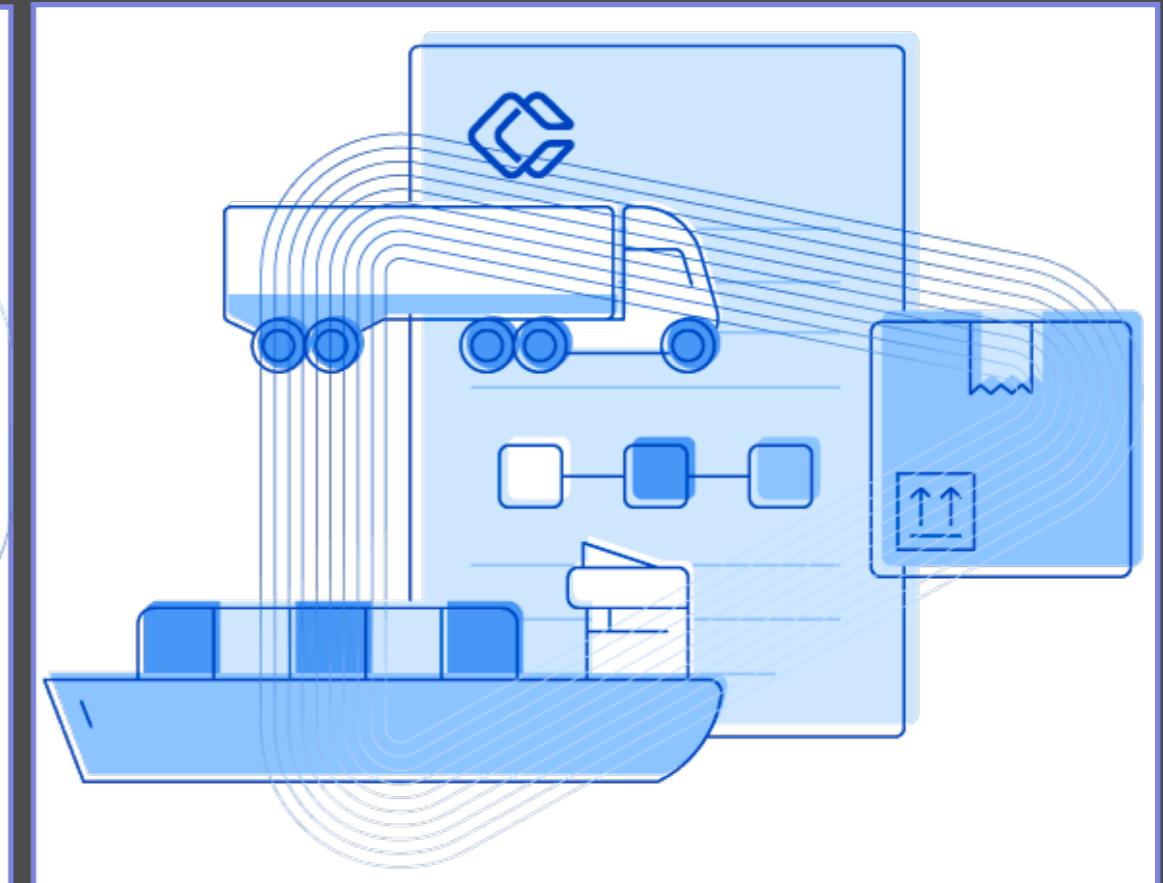
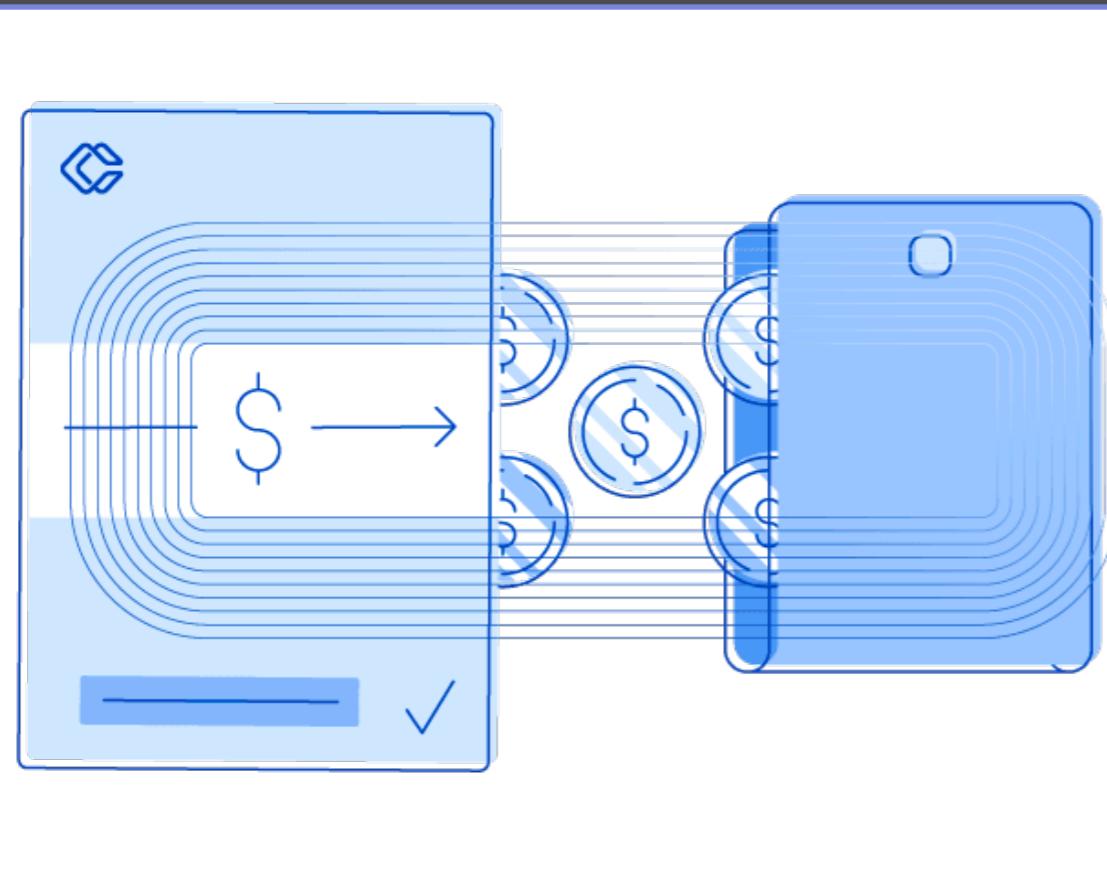
Logic: ERGO





CLAUSE

Platform connecting contracts to tools for automation





CLAUSE

Platform connecting contracts to tools for automation

The screenshot shows the Clause platform's user interface. On the left, there is a list of document templates represented by placeholder text boxes. On the right, there is a detailed view of a specific template, also showing placeholder text boxes. A large white arrow points from the Clause interface towards the DocuSign logo, indicating a connection or integration between the two platforms.

DocuSign®