

# 15-441: Computer Networks

## Recitation 3

P1 Lead TAs: Mingran Yang, Alex Bainbridge



# Agenda

---

1. Project 1 Checkpoint 3
2. Advanced Git
3. Q&A



# Checkpoint 3

*Due Sep 27, 2019*

This is only for **15-641** students!

You will need to

- daemonize your server
- Setup SSL/TLS
- Setup CGI

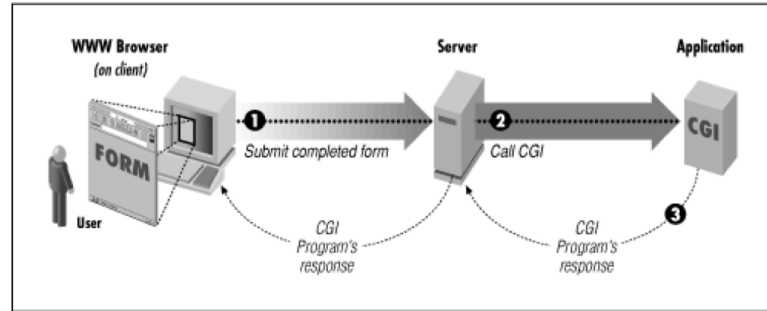
# Daemonization (Warning)

- **YOU NEED TO DO THIS FIRST**
- **NOT DOING THIS MAY RESULT IN YOUR GRADE BEING A “0”**
  - Many of our tests rely on your server being daemonized
  - We provide the skeleton to do it
  - All you need to do is add in some longjumps and resource management

# Daemonization (cont)

- Reminder: “rehashing a server” means you actually need to **restart** it and **reload** any configuration files
  - This is commonly used when changing things such as configuration files, SSL certificates, etc!
  - You should not start a new process
    - You should not just return either.
    - You need to do some work
  - Close your resources and open new ones.

# CGI: Generating Dynamic Content



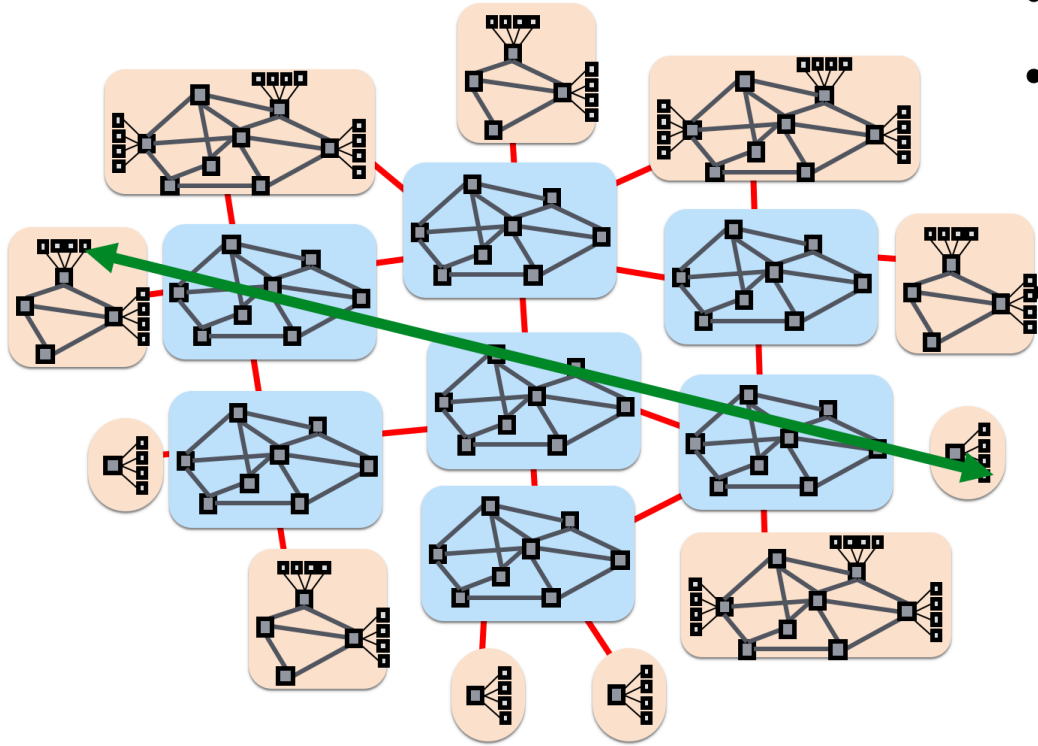
- Web server forward request plus additional information to an external application using a Common Gateway Interface
  - Where the user is connecting from, other user information
  - The CGI can access other data sources, e.g., databases
- CGI returns a response for the browser, e.g., HTTP document



# CGI For P1

- CGI allows for your server to become more responsive and interactive
- In this context you will use CGI to execute python scripts
  - You need to setup the python script for success by setting all the correct variables
  - Make sure that you aren't leaking memory or data.

# Transport Layer Security



- TLS secures the connection:
- Client authenticates server
- Needs the help from a Certificate Authority (CA)

Ensures integrity and confidentiality of the data

- Crypto magic – covered later in the course
- Transparent to application developers but not to the developer of Web server!

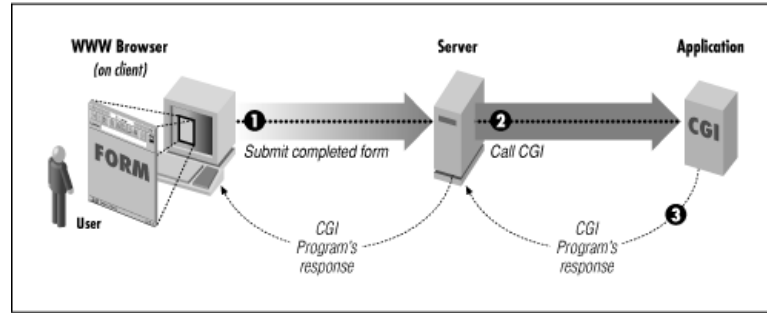




# SSL/TLS for P1

- You will need to get a certificate from “[project1.myheartisinthetwork.com](https://project1.myheartisinthetwork.com)”
- This will be used in the SSL library like the sample code we provide
- You will need to track which of your connections are https and which are http.

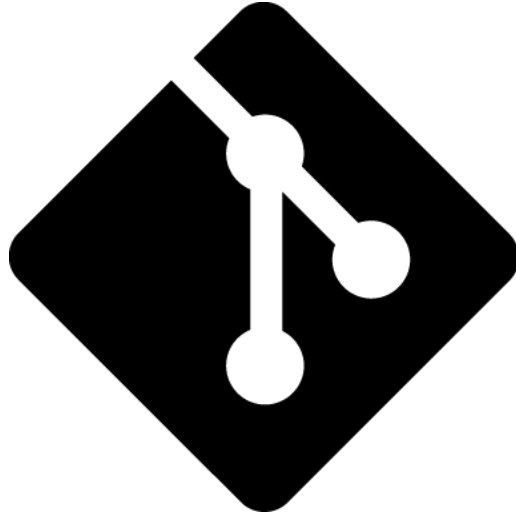
# CGI: Generating Dynamic Content



- Web server forward request plus additional information to an external application using a Common Gateway Interface
  - Where the user is connecting from, other user information
  - The CGI can access other data sources, e.g., databases
- CGI returns a response for the browser, e.g., HTTP document

Graphic: [https://www.oreilly.com/openbook/cgi/ch01\\_01.html](https://www.oreilly.com/openbook/cgi/ch01_01.html)

# Advanced Git



# Size Limits

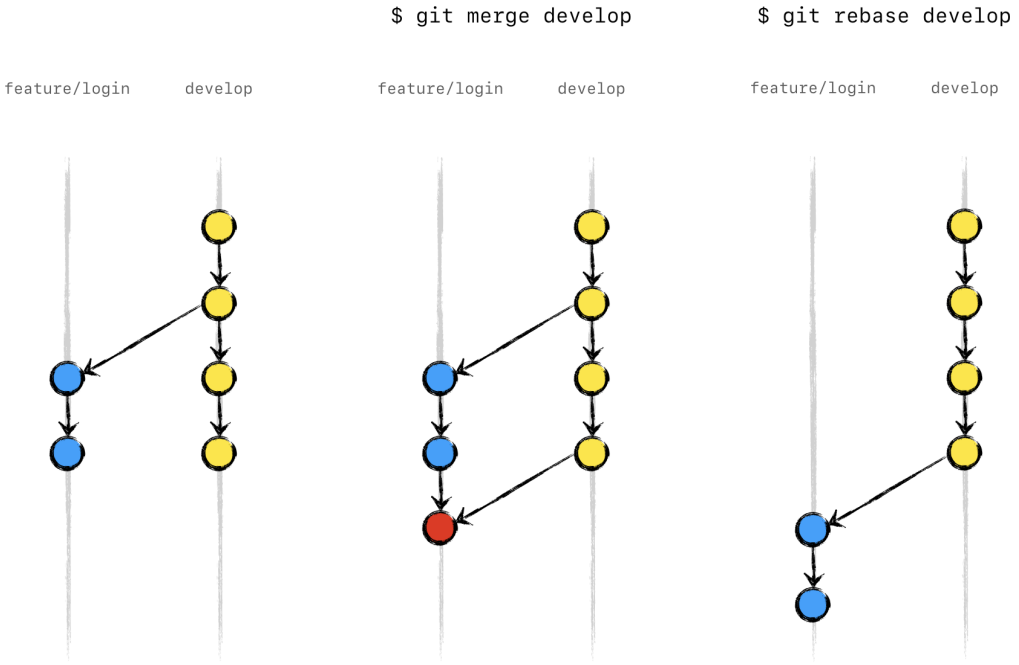
- Autolab allows maximum of 5mb per submission
- It is up to you to manage your size of your repository
- Size can increase drastically with
  - Git add .
  - Git add \*
- Make sure to
  - add only the necessary files
  - use git ignore for your object files – *Stated in the writeup as a requirement*

# Fixing a Bloated Repo

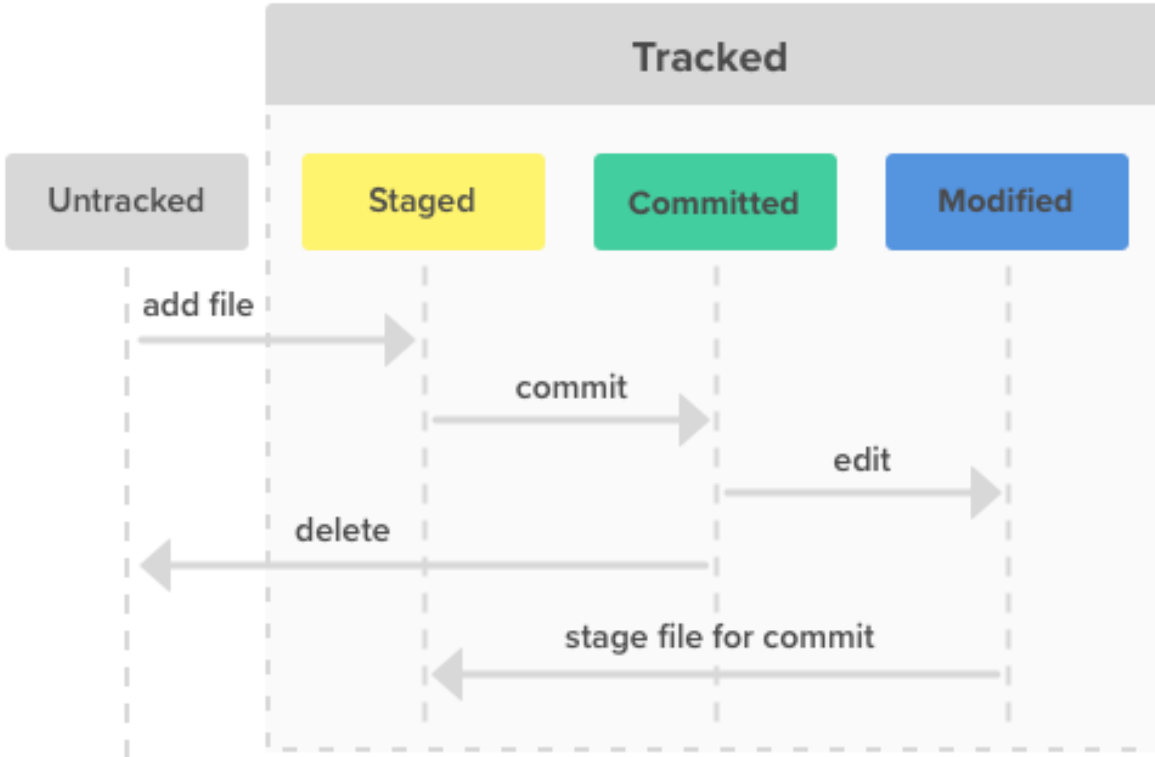
- Find your large files and commits
  - *Git rev-list [1]*
- Removing your cached files
  - *Git rm [2]*

# Git Rebase

- Applies commits onto a new starting point.
- Useful when features on one branch are desired on another branch.
- git rebase [3]



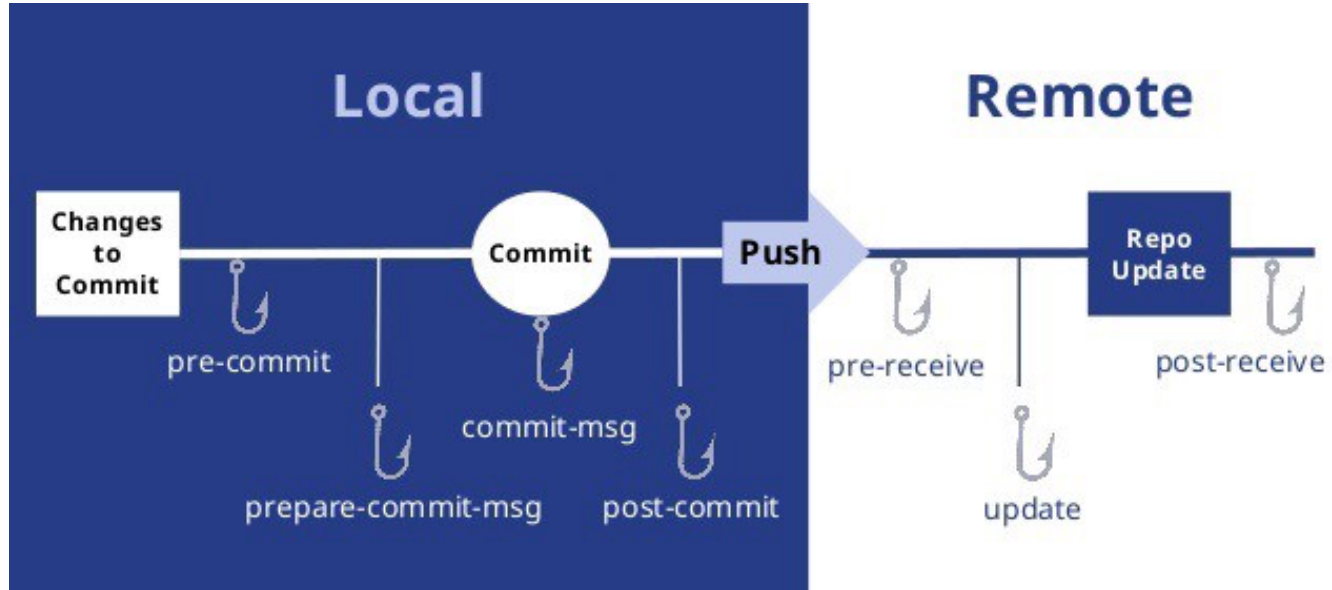
# Git Workflow



You all know this already

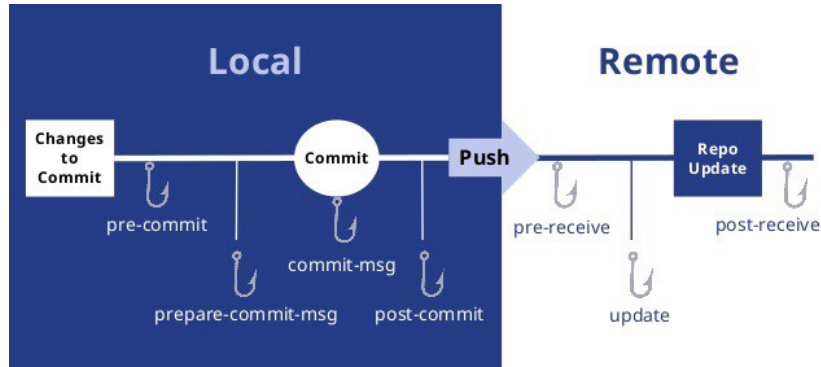
What if I told you there were hidden stages?

# Git Hooks



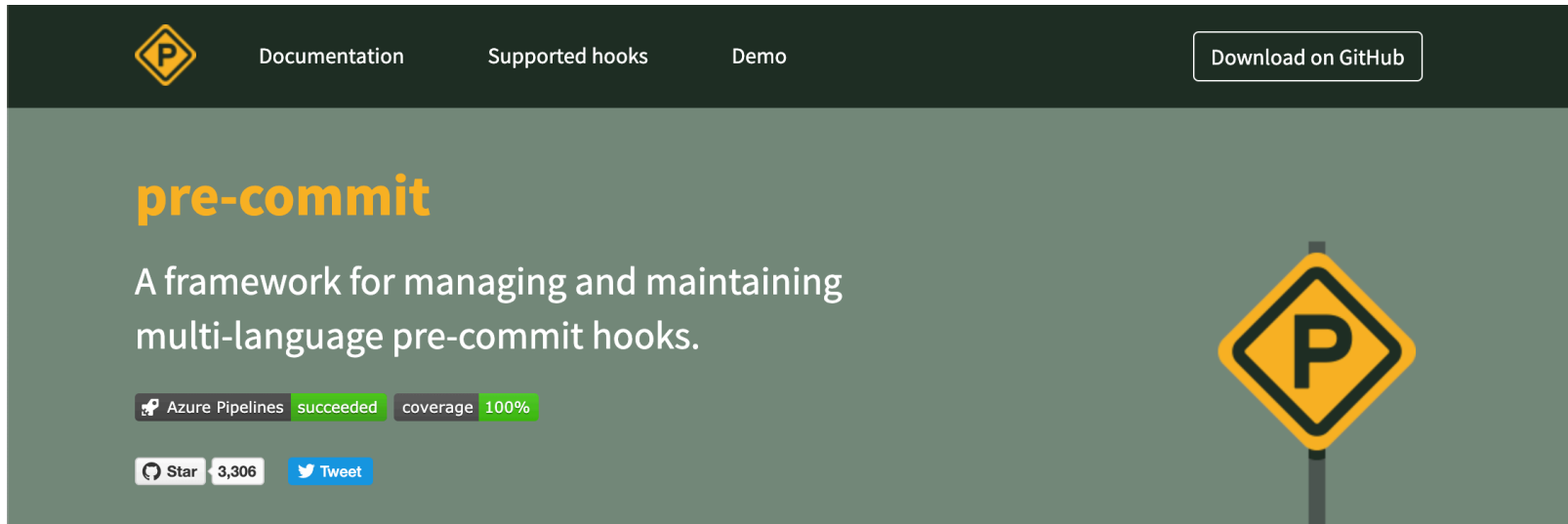


# Git Hooks



- Hooks allow for additional actions to be run at various points in the pipeline [4]
- There are three most common ones
  - Pre-commit
  - Pre-receive
  - Post-receive
- We will be focusing on Pre-commit

# Pre-Commit



The screenshot shows the homepage of the Pre-Commit project. At the top, there is a dark navigation bar with a yellow diamond logo containing a black 'P' on the left. To its right are the links 'Documentation', 'Supported hooks', and 'Demo'. On the far right of the navigation bar is a white button with a black border that says 'Download on GitHub'. Below the navigation bar, the main content area has a dark green background. On the left side, the word 'pre-commit' is written in a bold, orange, lowercase font. Below it, the text 'A framework for managing and maintaining multi-language pre-commit hooks.' is displayed in white. To the right of this text is a large, yellow diamond-shaped sign with a black border and a black 'P' in the center, mounted on a black post. Below the main text, there is a status bar with a small icon of a person, the text 'Azure Pipelines succeeded', and 'coverage 100%'. At the bottom left, there are two buttons: a white one with a star icon and the text 'Star 3,306', and a blue one with a bird icon and the text 'Tweet'.

- Pre-commit.com offers a framework as well as lots of tools that you can utilize for your projects [5]

# Pre-commit Capabilities

- Automatic Style Linting
  - You can also provide flags to tell it to fix your style automatically
  - \*Note: this isn't all you need to do for our style rubric. But at least you won't have to manually fix every line over 80 chars yourself 😊
  - Can run on more than just code, but also json and some data files!
- Library Management
- Test Suite Running
- Vulnerability Detection
- Static Code Analysis

# Pre-commit Setup

- Install pre-commit
- Create your config file
- Identify needs you want to address
- Find hooks that you will utilize
- Investigate their respective settings

# Pre-commit Setup

- Installation
  - `brew install pre-commit`
  - `pip install pre-commit`
- Check installation
  - `pre-commit --version`
- Create `“.pre-commit-config.yaml”` file in top level directory
  - Add in the hooks and settings desired

# Sample YAML

repos:

- repo: <https://github.com/pre-commit/pre-commit-hooks>  
rev: v2.3.0  
hooks:
  - id: check-yaml
  - id: end-of-file-fixer
  - id: trailing-whitespace
- repo: <https://github.com/psf/black>  
rev: 19.3b0  
hooks:
  - id: black

# Sample YAML Explained

- The repo: <https://github.com/pre-commit/pre-commit-hooks> has 3 hooks we will be using
- The repo: <https://github.com/psf/black> has 1 hook we will be using
- None of the hooks have special settings activated

# Finding Hooks

- Hooks will be in Github, and most of them can be found through Google search
- Here's a good list to get started : <https://pre-commit.com/hooks.html>
  - You may need to check on a given hook which languages it supports.
  - Some are language dependent some aren't



Q & A



# References

1. <https://stackoverflow.com/questions/10622179/how-to-find-identify-large-commits-in-git-history>
2. <https://help.github.com/en/articles/removing-files-from-a-repositorys-history>
3. <https://git-scm.com/docs/git-rebase>
4. <https://medium.com/@suthagar23/git-hooks-keep-the-code-quality-119e6feb511e>
5. <https://pre-commit.com/>