Visualizing Git/Github

CS524: Project Proposal

Yash Kurkure

Challenges

- Visualizing a code base as a physical map.
- Modular codebases and contributions.
- Evolution of the code base over time.

Existing Works

- **git log**

```
$ TZ=PST8PDT git log-compact --decorate --graph -n 17 v2.6.1
  === 2015-09-28 ===
* 22f698cb 19:19 jch (tag: v2.6.1) Git 2.6.1
*   3adc4ec7 19:16 jch Sync with v2.5.4
|\
| * 24358560 15:34 jch (tag: v2.5.4) Git 2.5.4
| *   11a458be 15:33 jch Sync with 2.4.10
| |\
| | * a2558fb8 15:30 jch (tag: v2.4.10) Git 2.4.10
| | *   6343e2f6 15:28 jch Sync with 2.3.10
| | |\
| | | * 18b58f70 15:26 jch (tag: v2.3.10, maint-2.3) Git 2.3.10
| | | *   92cdfd21 14:59 jch Merge branch 'jk/xdiff-memory-limits' into maint-2.3
| | | |\
| | | | * 83c4d380 14:58 jk  merge-file: enforce MAX_XDIFF_SIZE on incoming files
| | | | * dcd1742e 14:57 jk  xdiff: reject files larger than ~1GB
| | | | * 3efb9880 14:57 jk  react to errors in xdi_diff
| | | * |   f2df3104 14:46 jch Merge branch 'jk/transfer-limit-redirection' into maint-2.3
| | | |\ \
| | | | | | === 2015-09-25 ===
| | | | * | b2581164 15:32 bb  http: limit redirection depth
| | | | * | f4113cac 15:30 bb  http: limit redirection to protocol-whitelist
| | | | * | 5088d3b3 15:28 jk  transport: refactor protocol whitelist code
| | | | | |   === 2015-09-28 ===
| | | * | |   df37727a 14:33 jch Merge branch 'jk/transfer-limit-protocol' into maint-2.3
| | | |\ \ \
| | | | |/ /
| | | | | /
| | | | |/
| | | |/|
| | | | | === 2015-09-23 ===
| | | | * 33cfccbb 11:35 jk  submodule: allow only certain protocols for submodule fetches
```

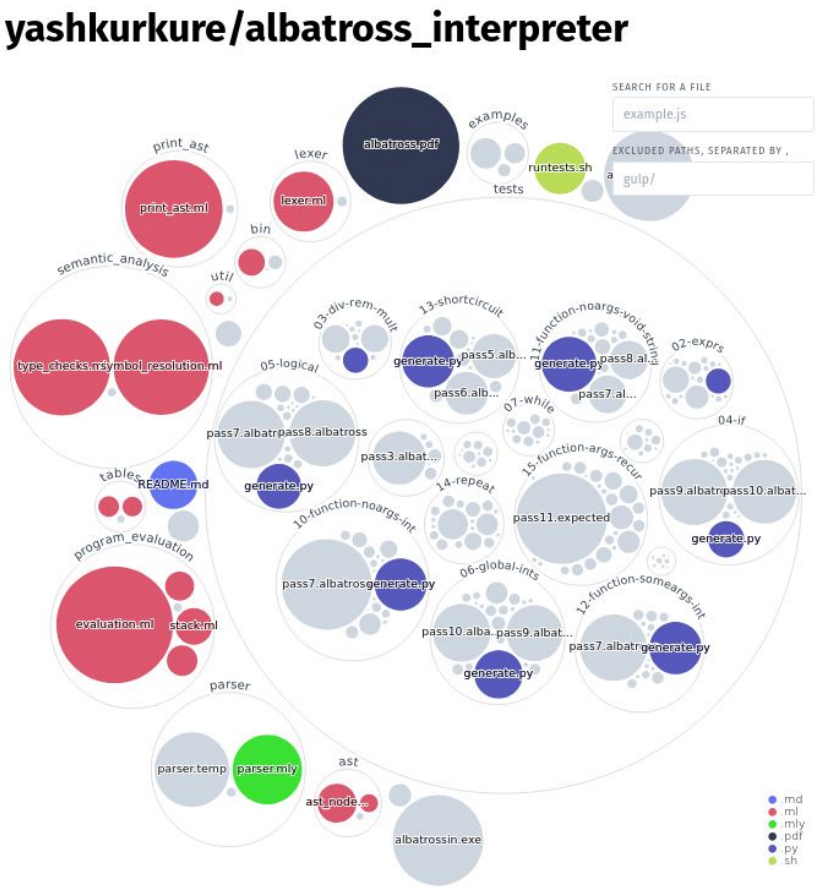
Existing works

Visualizing a code base as commits are made to it.

Graphical visualisation of all Ethereum Github repositories from 2013 to 2018.

Existing works

https://githubnext.com/projects/repo-visualization/



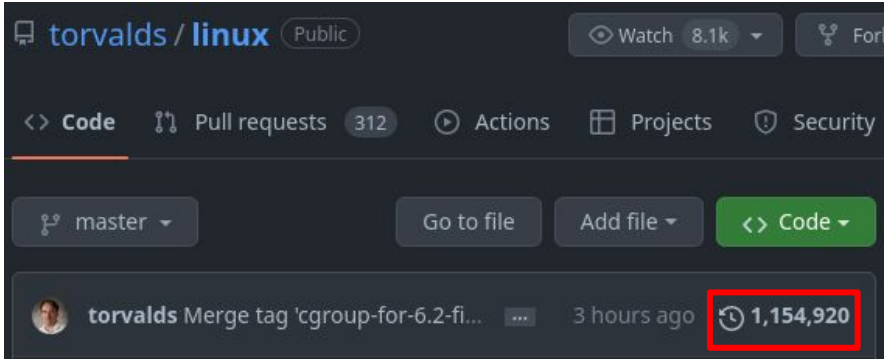
The DataSet

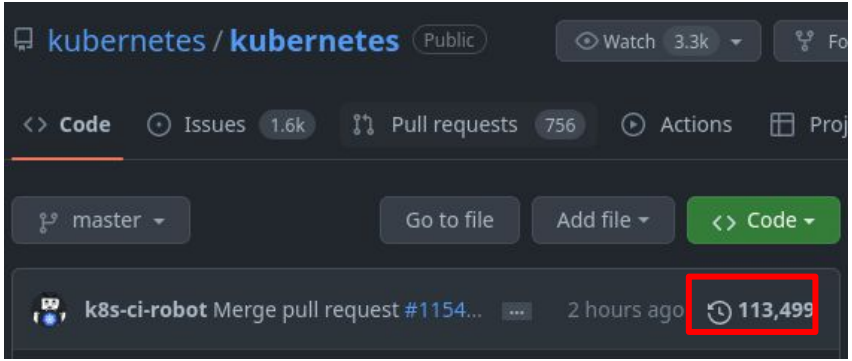
Open source projects hosted on github

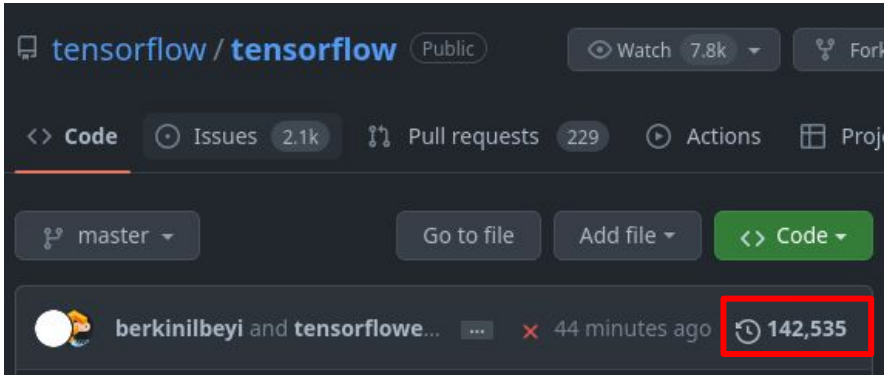
The goal is for the application to produce a visualization of an arbitrary git repository.

The DataSet

Git repositories with enough amount of data?







The DataSet

But what is interesting about them?

There are many things you can visualize using a git project or any source code.

- The branch/commit tree
- Pull requests
- The project structure
- Tracking file progression

Example: Scraping Data from git log

Running this in a cloned git repository locally:

```
git log | sed -nr '/Author:/p' log | awk -F '@' '{print substr($2, 1, length($2)-1)}' | sort | uniq -c | sort -bgr
```

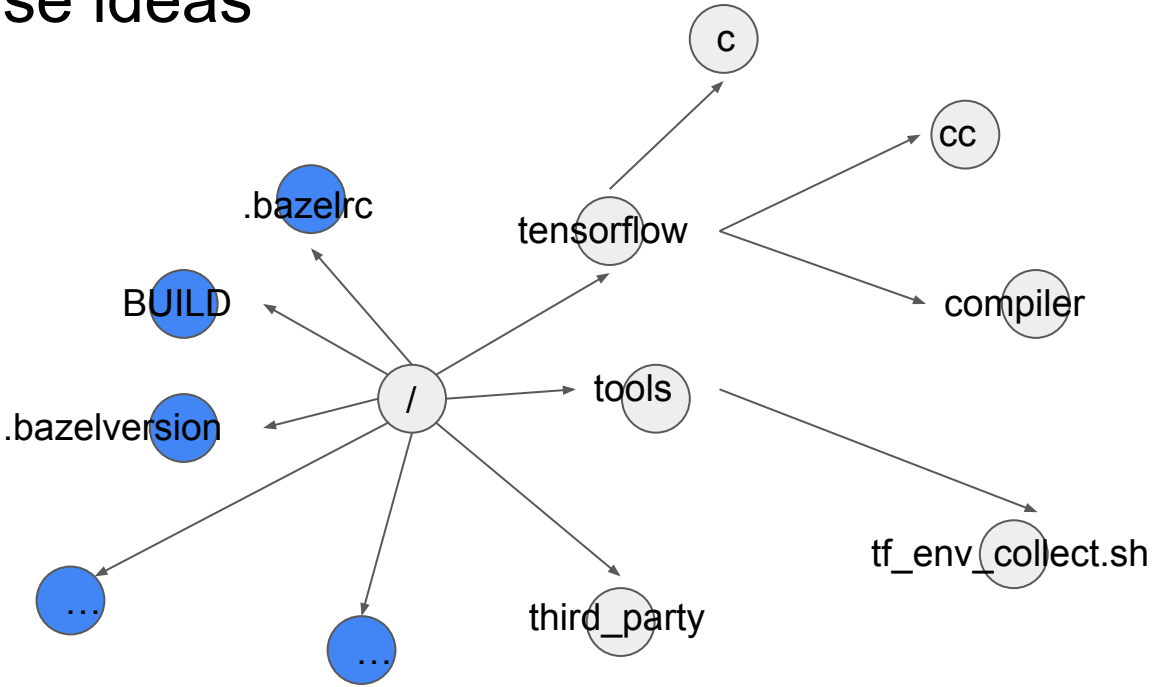
```
72680 google.com
42402 tensorflow.org
 9834 gmail.com
 3386 users.noreply.github.com
 2927 nvidia.com
 2236 intel.com
 1508 outlook.com
  824 arm.com
  431 huawei.com
  374 us.ibm.com
  341 graphcore.ai
  313 amd.com
  296 in.ibm.com
  249 ibm.com
  213 hotmail.com
  179 bdti.com
  127 microsoft.com
  120 amazon.com
  117 qq.com
  116 ceva-dsp.com
  109 codeplay.com
```

Company contributors?

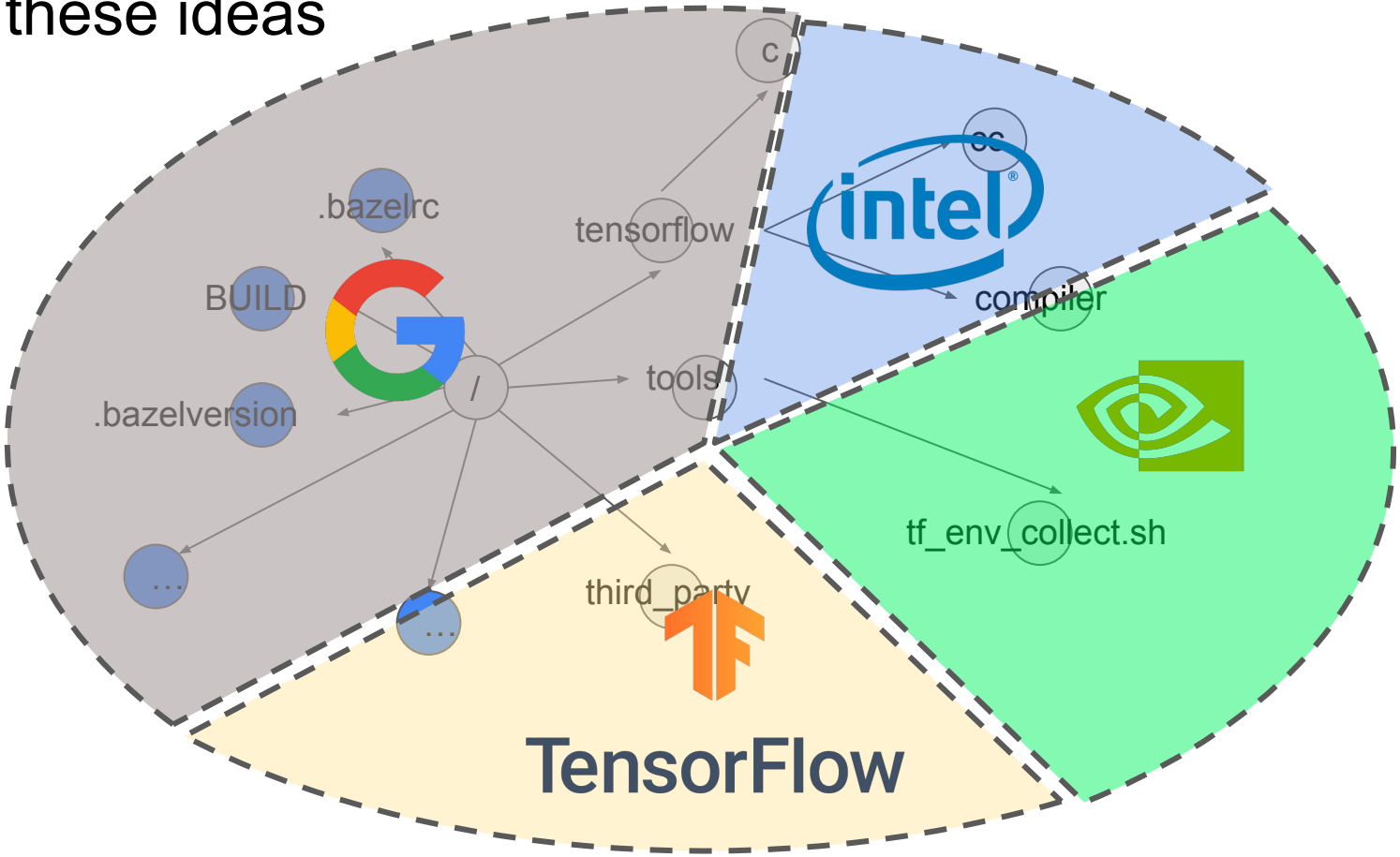
Independent Contributors?

Combining these ideas

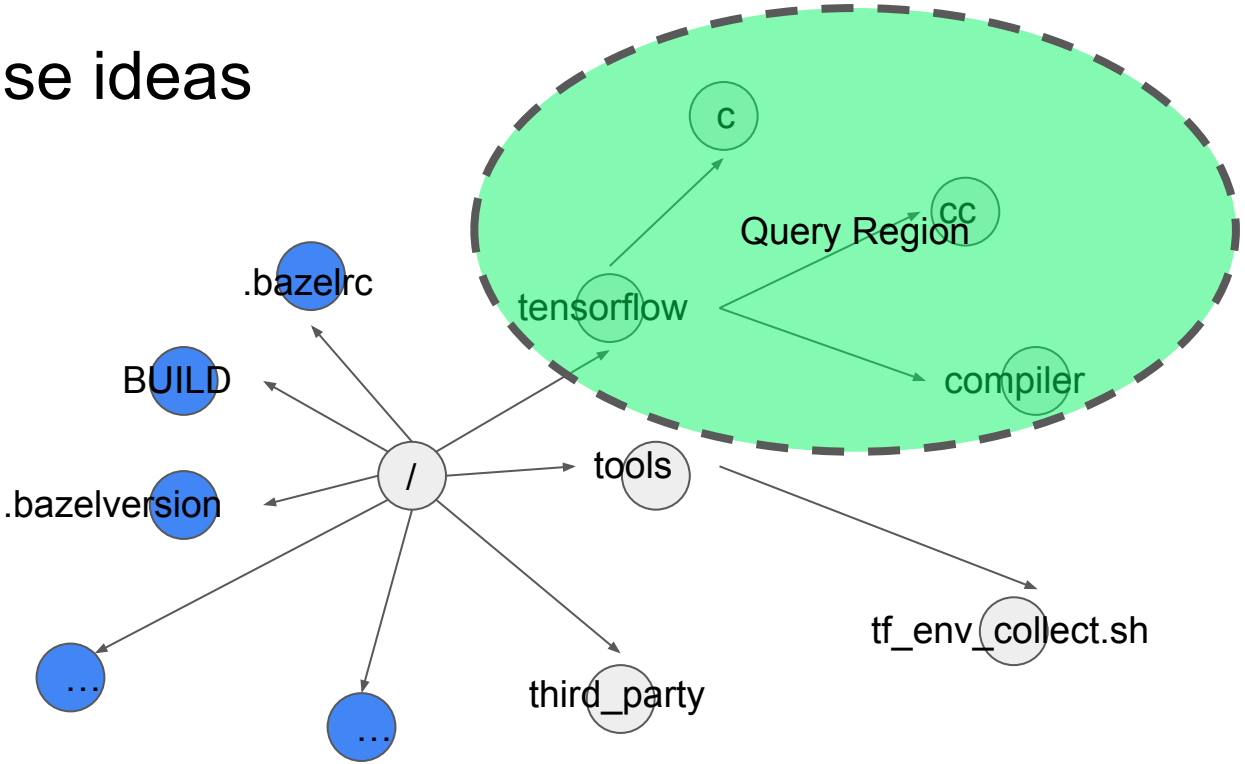
Combining these ideas



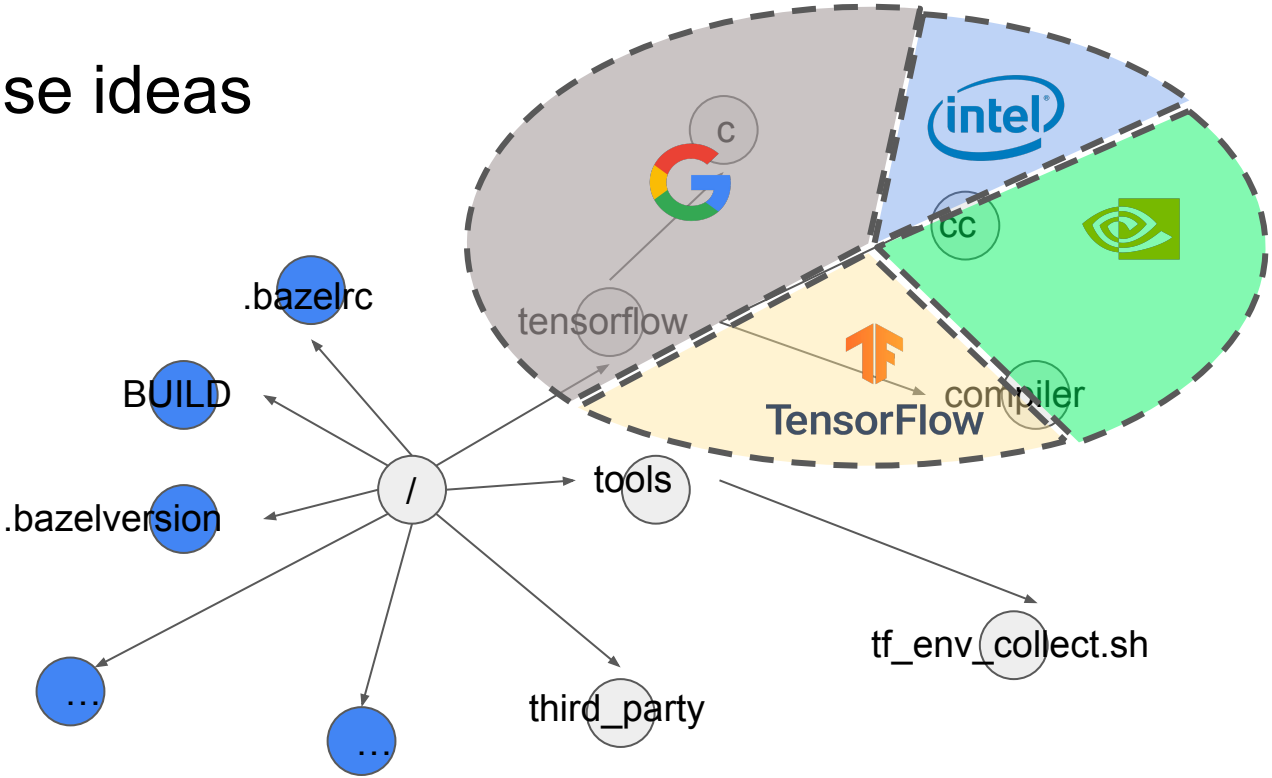
Combining these ideas



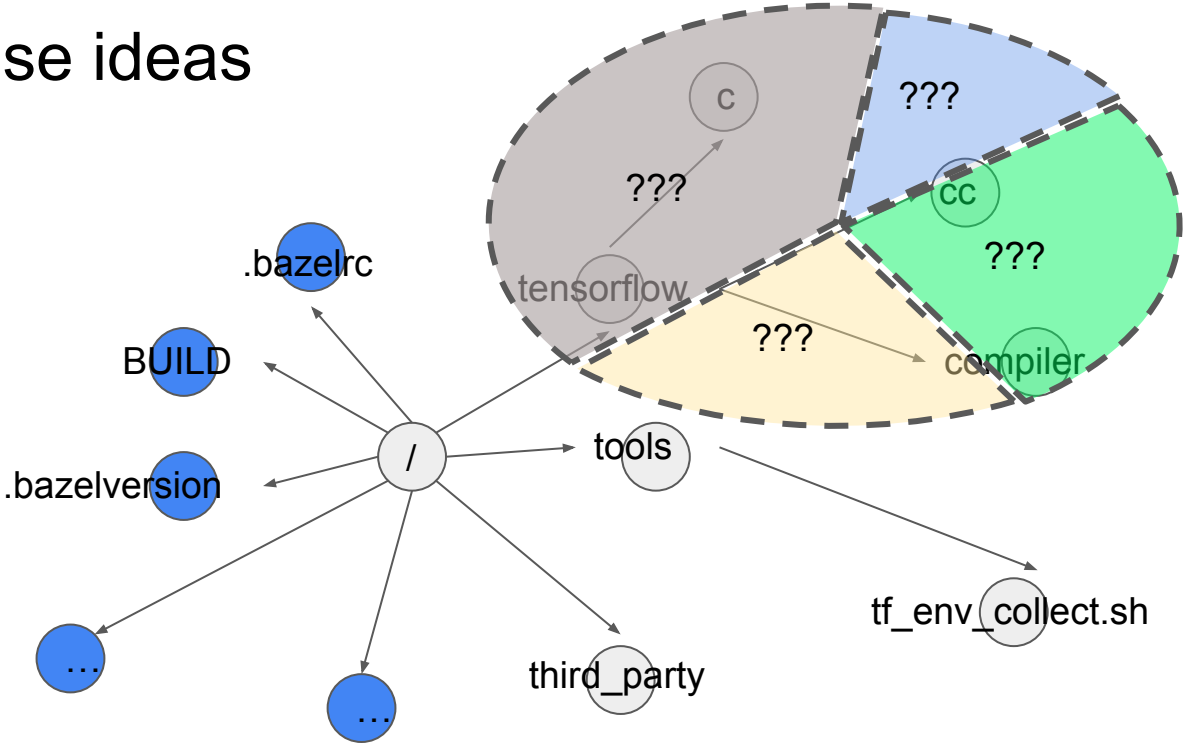
Combining these ideas



Combining these ideas



Combining these ideas

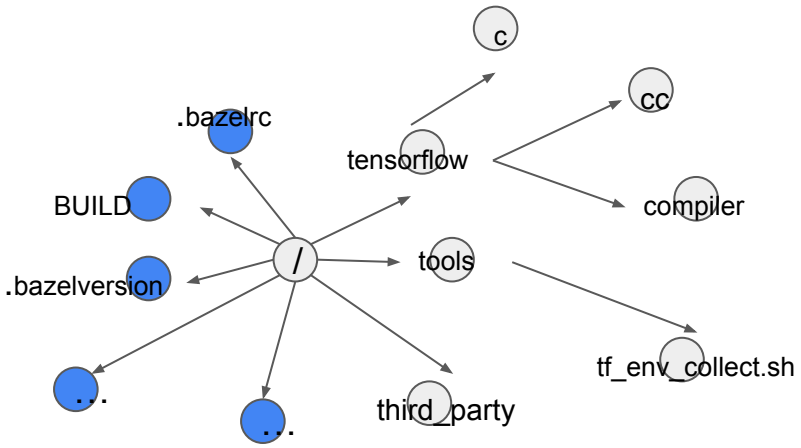


Visualization Areas: Representation & Interaction

- Representation & Interaction
  - This area focuses on the design of visual representations and interaction techniques for different types of data, users, and visualization tasks.
  - Eg: Visualizing a **directorial structure as a tree**.

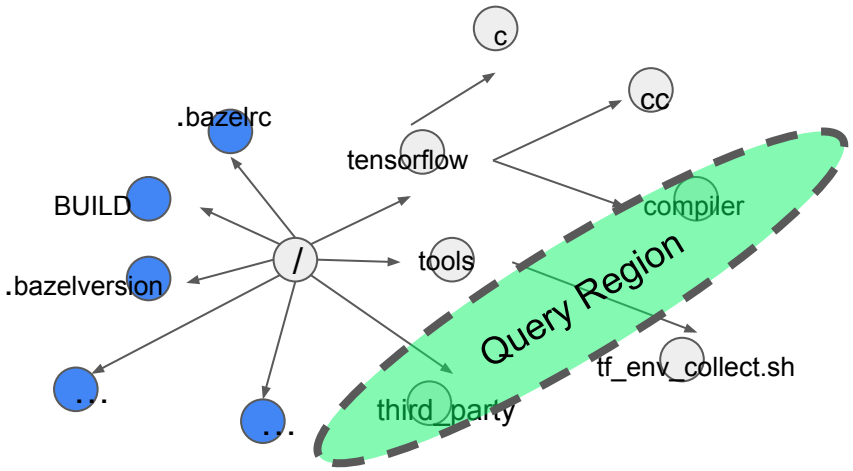
Visualization Areas 1: Representation & Interaction

- **Representation** & Interaction
  - This area focuses on the design of visual representations and interaction techniques for different types of data, users, and visualization tasks.
  - Eg: Visualizing a **directorial structure as a tree**.



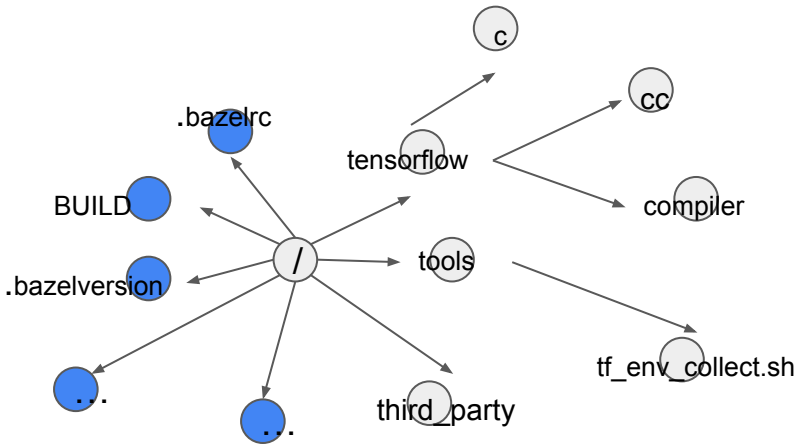
Visualization Areas 1: Representation & Interaction

- Representation & **Interaction**
  - This area focuses on the design of visual representations and interaction techniques for different types of data, users, and visualization tasks.
  - Eg: Visualizing a **directorial structure as a tree**.



Visualization Areas 2: Data Transformation

- Data Transformation
  - This area focuses on the **algorithms and techniques that transform data from one form to another to enable effective and efficient visual mapping** as required by the intended visual representations.
  - Eg: Querying for the **list** of commits over a code base **tree**

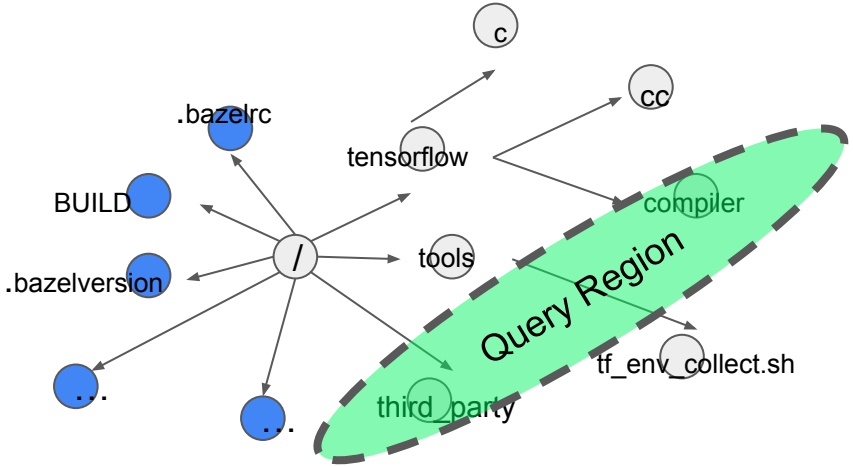


Visualization Areas 2: Data Transformation

Give me the **list of commits/incoming pull requests** in the **selected query region**.

~~Iterate through all commits?~~

Pre process the data



Milestones

- **Scraping Data** from GitHub and /.git
- **Store data** effectively to facilitate queries.
- **Implement simple queries** first (e.g: “List all commits in query region”)
- **Implement complex queries** - (a combination of simple queries? E.g: Company contributions -> list commits + filter for certain email domains)
- **Front end** development.

Milestones - Weekly Plan

Week(s)	Task 1	Task 2
5	**Project Proposal**	Data Scraping Scripts
6	Data Scraping Scripts	Formulate interesting queries
7-9	Data storage	Run queries through command line
10	**Midterm Review**	Visualization Front End
10-11	Visualization Frontend	More complex queries
12-13	Visualization Frontend	Deploy to Github Pages?
14	Project Report + Docs	Project Presentation
15	**Final Review**	