

# Get Higher Productivity and Smoother Tape-out by Smarter Disk Space Management

- Jigar Savla -  
{ jigar.savla@gatech.edu }

## Introduction

Q: Are we making best use of our time solving Disk Space issues rather than solving hard Silicon problems?

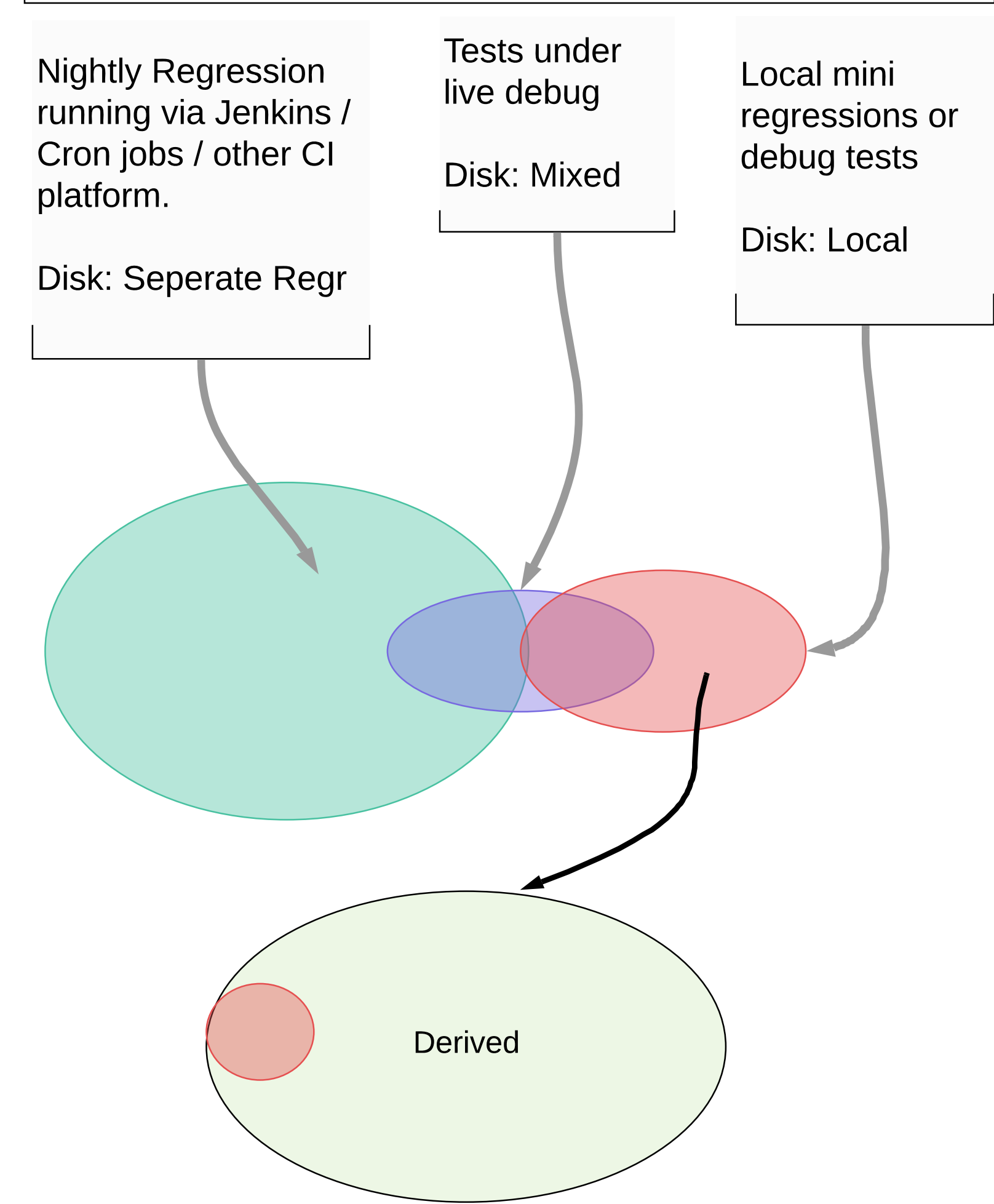


Figure: Actual usage, the red circle at the bottom indicates non-derived file usage

"derived": generated files like dumps, logs which can be re-created and backup copies "non-derived": like source code

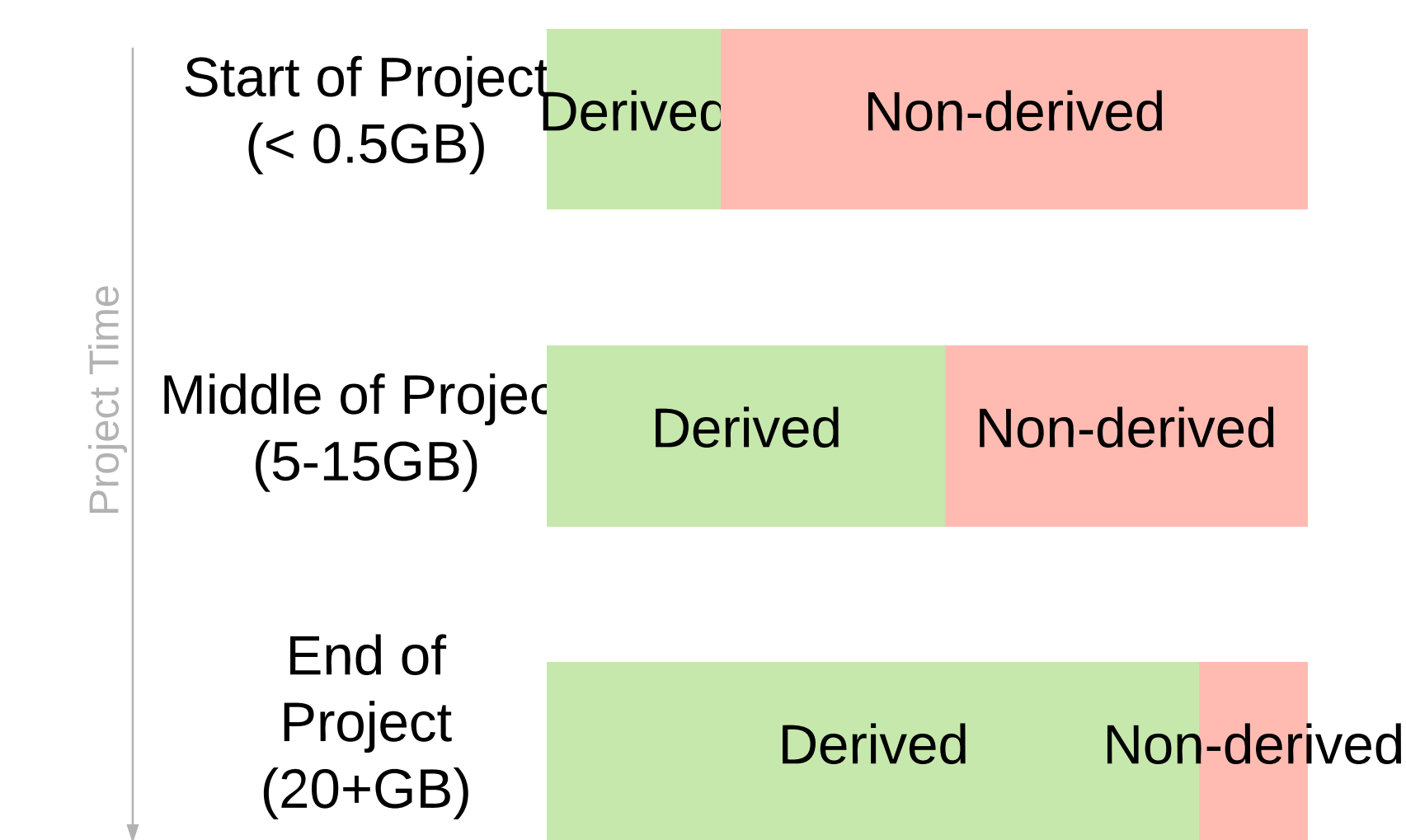


Figure: Disk space sharing %age between derived and non-derived files

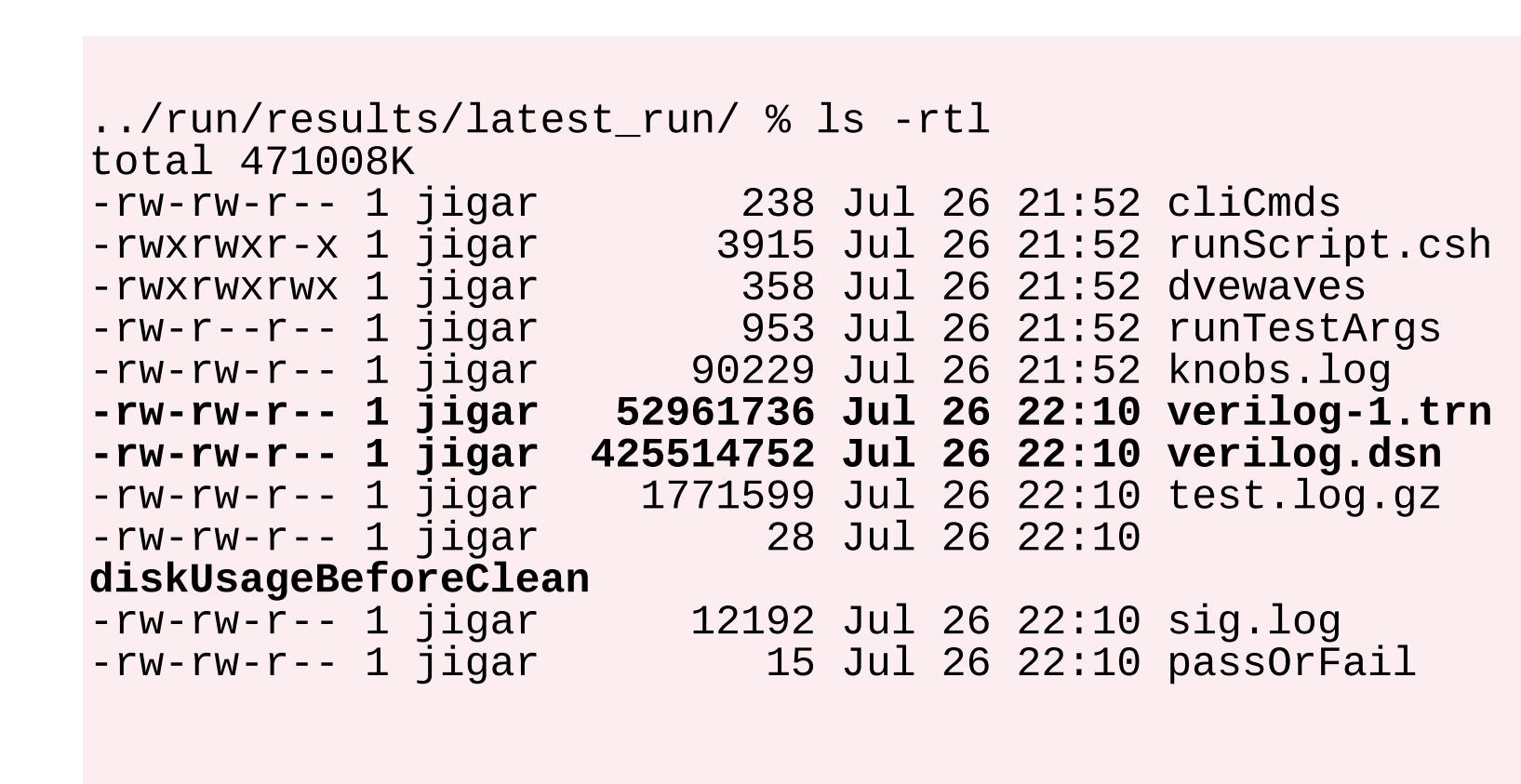


Figure: Typical run dir with waveform dumps

### EXISTING SOLUTIONS:

- Jenkins / Continuous Integration Continuous Delivery (CI/CD)
- Turn off Backup snapshot of the disks for regression disks
- Use compression for the log files.
- Using the native Disk quotas in inodes of Unix filesystem.
- Most typical is running, 'du' once the disk reaches 90% capacity.
  - Problem with this is:
    - How does the user know which directory is the biggest one, how old it is? When was it last accessed?

## Methods

**Our Goal:** Give users easy to use tools and simplify the user experience

### Our approach:

- Idea: Seperate out *derived* files from the *non-derived* / *source* files.
- Why?:
  - Disk space is filling out irrespective of disk space thrown at users
  - We have seen build, test results i.e. **derived** files occupy most of the disk space
- Root problem:
  - Analyzing current breakup of files : **70%+** consumed by derived files
  - We chose the 'aging' approach. We once again found through gut intuition and backed up by statistical analysis that anything more than two months old is **97.4% guaranteed** to not be used again.

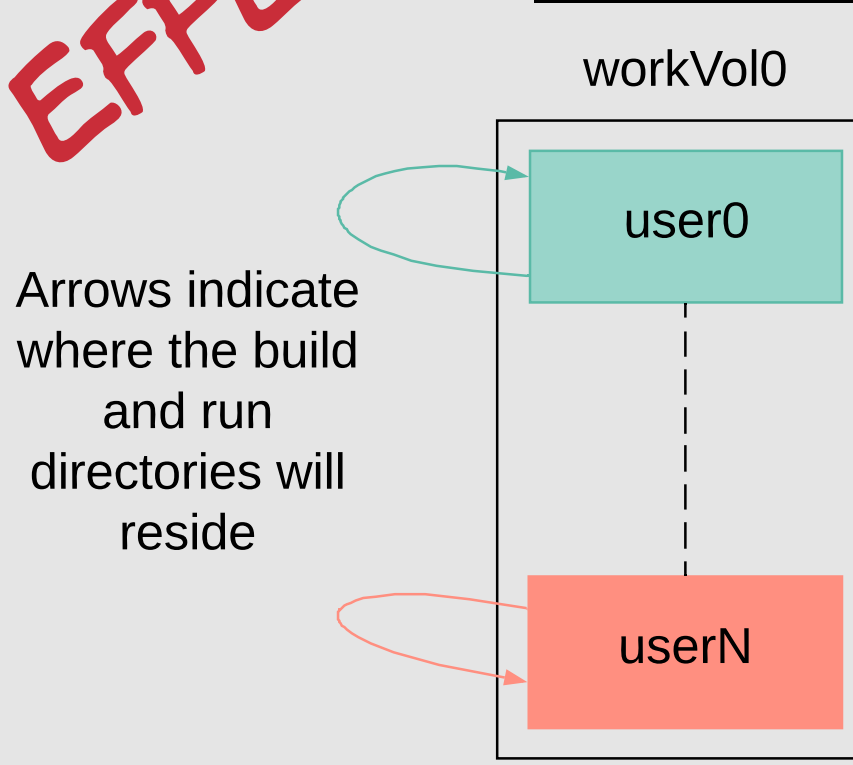
```
From: automated_email_server
Sent: today
To: high_consumption_users@coolcompany.com
Subject: Disk Usage Report
```

```
/vol3 is 88% Full
8.2T / 7.8T Used, 400G Available
```

```
Size Directory
467G /vol3/users/alpha
362G /vol3/users/beta
355G /vol3/users/charlie
...
171G /vol3/users/zulu
```

**NOT EFFECTIVE**

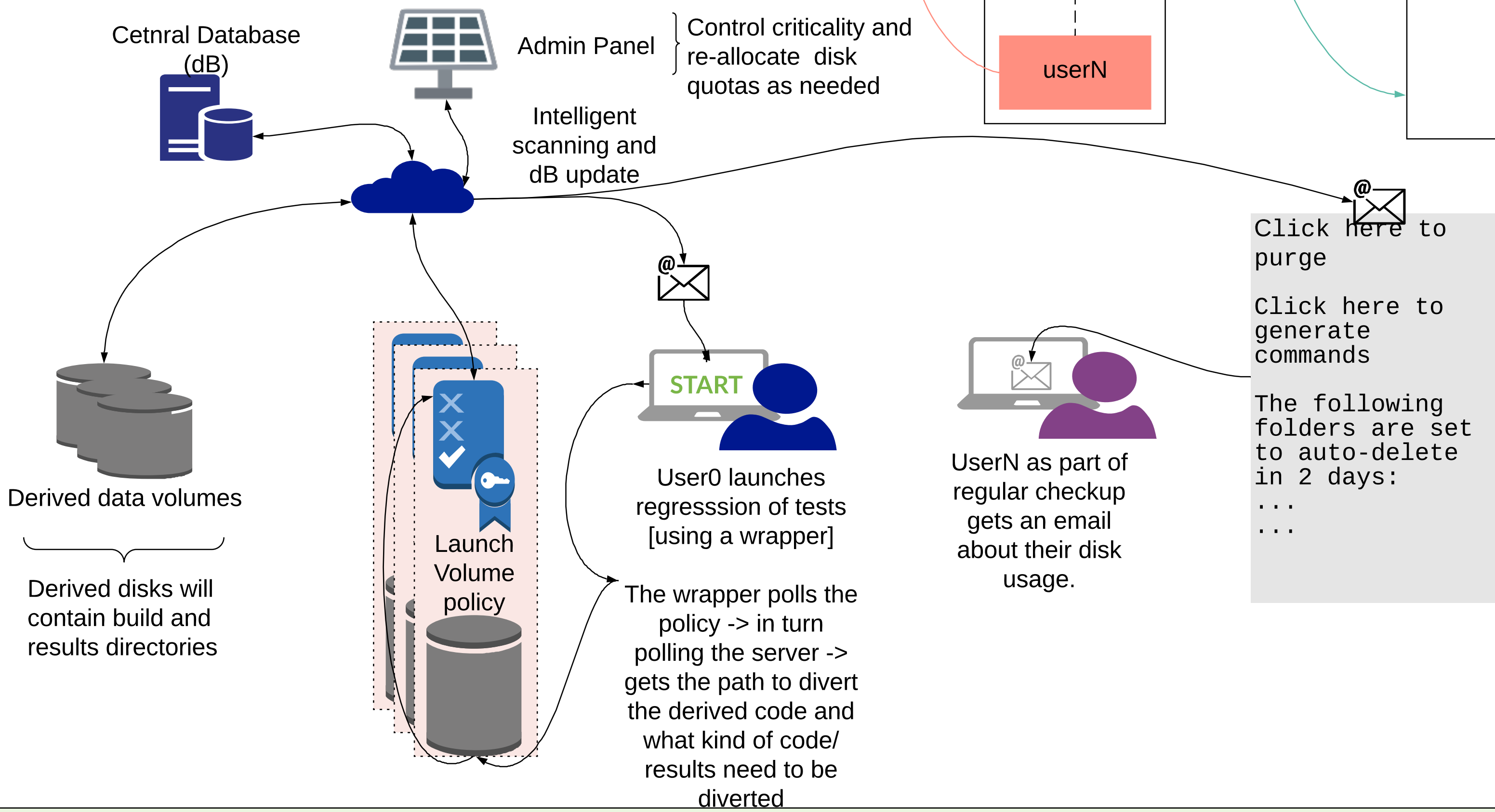
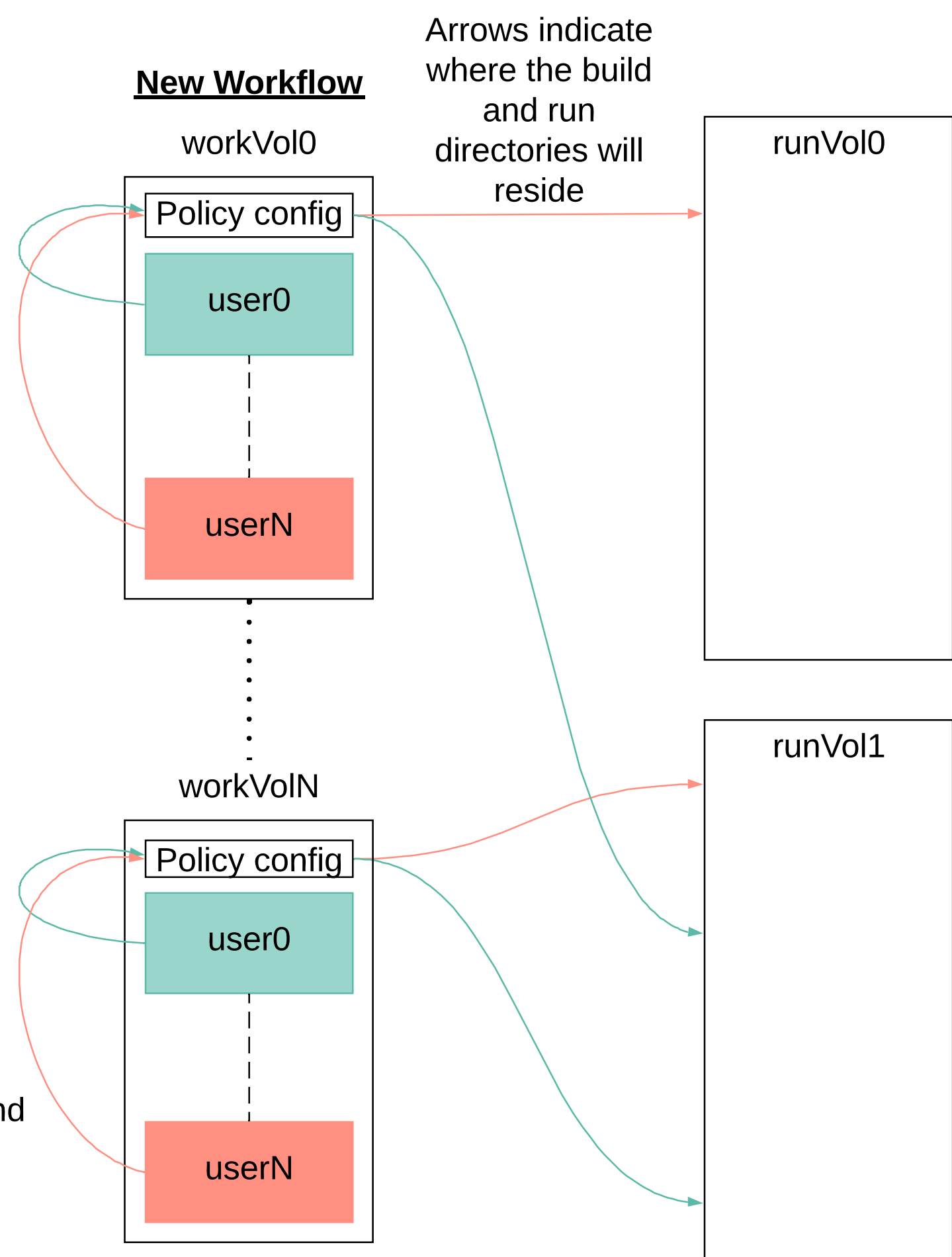
### Old Workflow



### Solution Code Structure:

- Policy based approach – for easier administration
  - 'Policy' files are config files per disk volume to indicate which volumes a user's build directories should get routed to. Target volumes unique to each user.
- Database design
  - Three relational databases
    - Owner table
    - RunDir table
    - Run table
- Automated email notification procedure
  - Daily routine
- Open Source code to leverage
  - Go's concurrency support was a very valuable scaling tool
- Quick summary
  - To keep the existing workflow mostly intact, we provide the ability to redirect storage of derived files to a different volume while keeping the work area in the same volume

### New Workflow



## Conclusion

Code will be available soon on: [github.com/jigarsavla](https://github.com/jigarsavla)



- We provide the ability to redirect storage of derived files to a different volume while keeping the work area in the same volume; nothing changes, not even the sandbox structure nor the work flow, just the location of derived files.
- Use this structure to automate cleaning of derived files; after not touching the files for a specified amount of time, notify the user of their largest sized run directories and provide remove commands to copy and paste into their terminal.
- Simplifying the user experience lead to larger adoption.
- Any time lost due to disk space exhaustion means delayed time to market, which means lost revenue (\$\$).