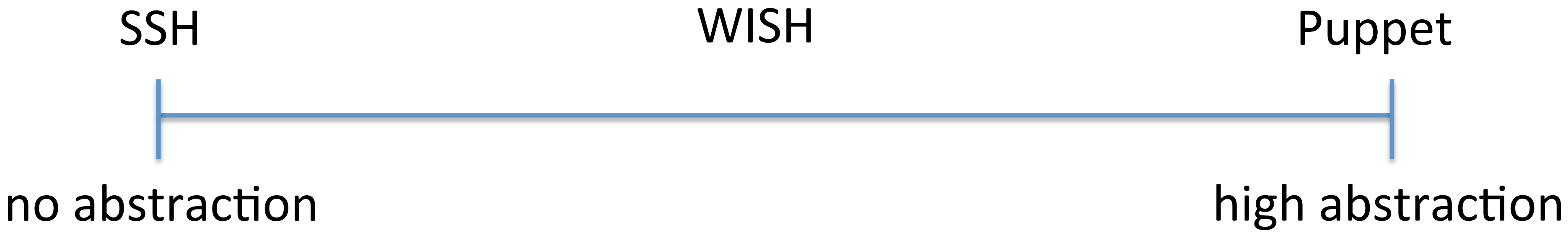
WISH:
the Wide-area Interactive SHell

Nick Jones, Jude Nelson, and Arman Suleimenov

WISH Motivation

- Shell scripting is *simple* and *popular*
- Lots of people need to manage systems
  - Shell scripting alone is insufficient
- Existing tools:
  - Puppet (et al.): scalable, no shell semantics
  - Parallel SSH: shell semantics, not scalable



WISH Goals

- Extend shell programming into the wide area
- Provide standard distributed programming primitives:
  - Spawn, Join, Synchronize, and Signal
  - Inter-process and inter-host communication
  - Globally accessible processes, files, and environment variables

WISH Implementation

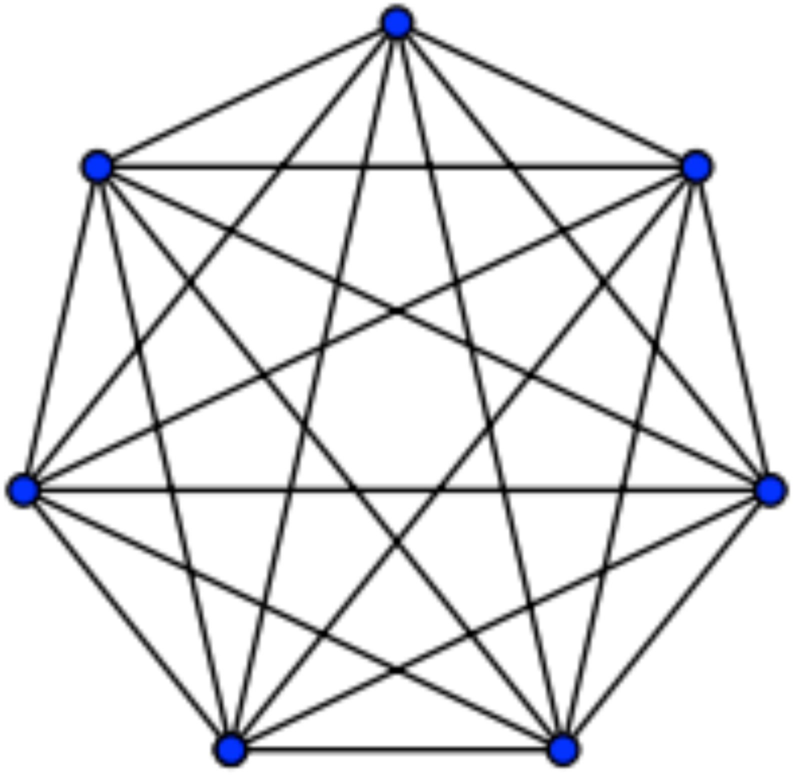
- WISH components:
  - WISH Daemon
  - WISH Client
  - WISH Command line tools

WISH Federation

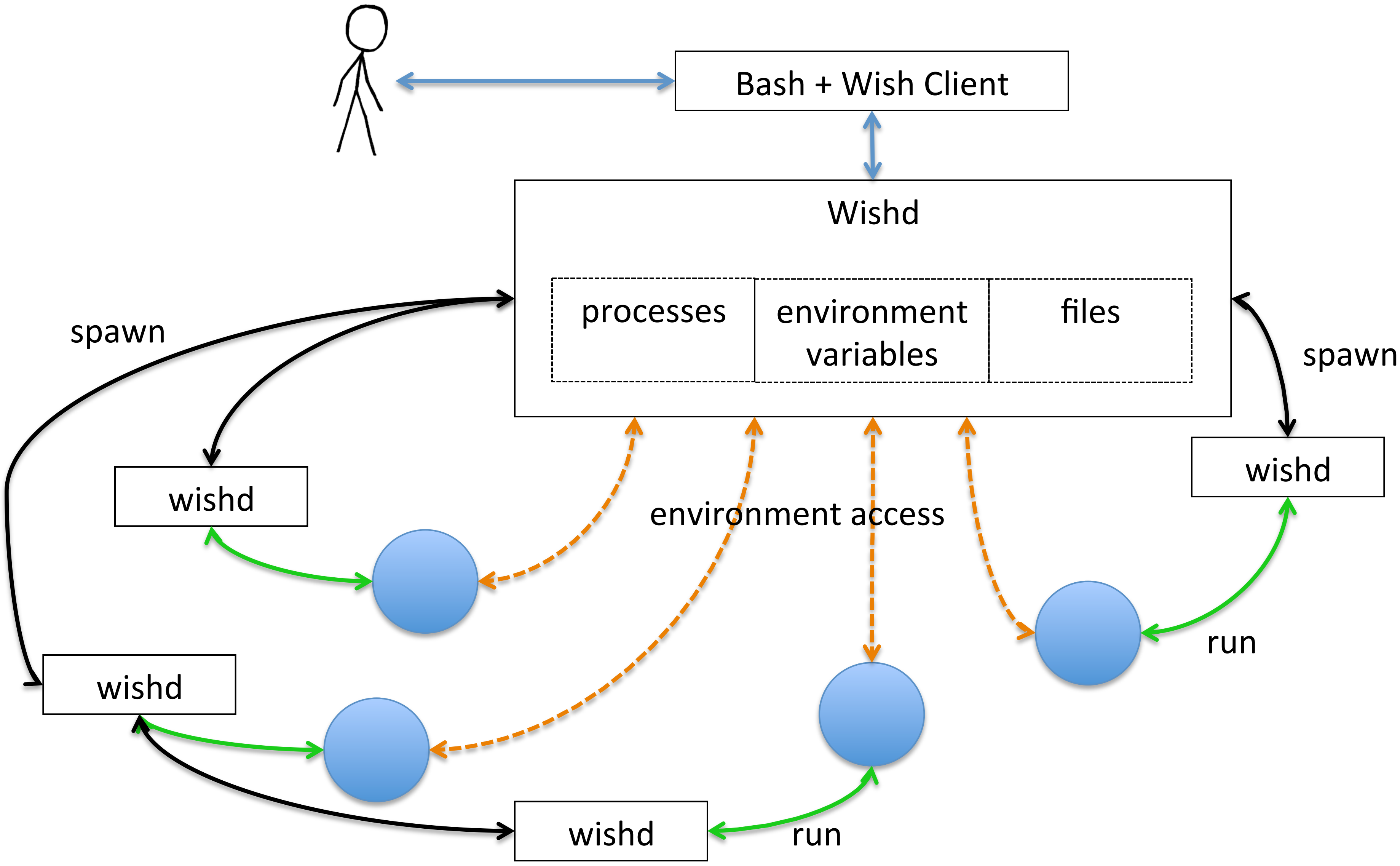
- Each WISH node maintains a connection to every other WISH node

- Nodes monitor each other's state

- Nodes initially know of all other nodes



WISH Architecture



Process Lifecycle

1. Origin: wishd receives job from localhost
2. Origin: wishd sends job to remote wishd
3. Remote: daemon forks child to run job
4. Remote: child launches process, sends global PID to originator
5. Remote: child streams output to originator
6. Remote: child waits for process completion
7. Remote: sends exit code to originator

Processes

- Ability to launch, join, and signal with any process on any node within WISH
- Globally unique PIDs

```
pspawn –c “echo HELLO” node5.princeton.vicci.org

pjoin GPID

psync GPID1 GPID2 GPID3…

psig –9 GPID
```

Files

- Ability to direct input and output streams from processes running on any host

```
pspawn –i hello.txt –o goodbye.txt –c “grep macaroni”  node8
```

Files

- WISH can globally expose files to other hosts within the WISH federation
- Accessible via globally unique HTTP URL
  - http://node5.princeton.vicci.org/FILE/data.tar.gz
  - $FILES_ROOT/data.tar.gz
- Per node access control

```
    fshow myfile.txt

    fhide passwords.txt
```

Environment Variables

- Globally visible environment variables on all hosts
- Atomically test-and-set environment variables

```
gsetenv WISHVAR 1

ggetenv WISHVAR

taset WISHVAR false true
```

Resource Monitoring

- Ability to rank hosts by resource utilization
- Returns a value easy to script with

```
nget –l 1

nget –r 1
               pspawn –c COMMAND $(nget –l 1)
nget –c 1

nget –d 1
```

WISH Client

- Interactively view output from remote WISH commands

```
bash-3.2# ./wlistend
wlistend daemon starting up
starting the daemonizing process
```

WISH Demo

- Process watchdog service
- Generation of SSL Certificate Signing Requests

Future Work

- Security
- nget boolean expressions
- Gossiping protocol for node discovery
- Global “ps” command
- join-all

Conclusions

- WISH extends shell programming into the wide area
- Provides tools for job launching, synchronization, communication, and management