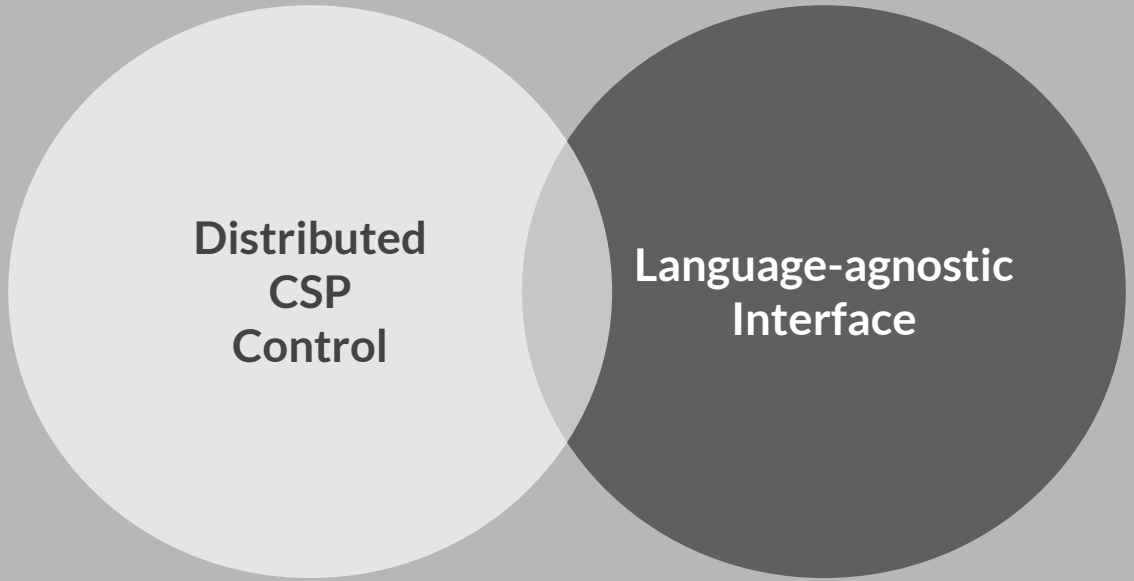
The Circuit

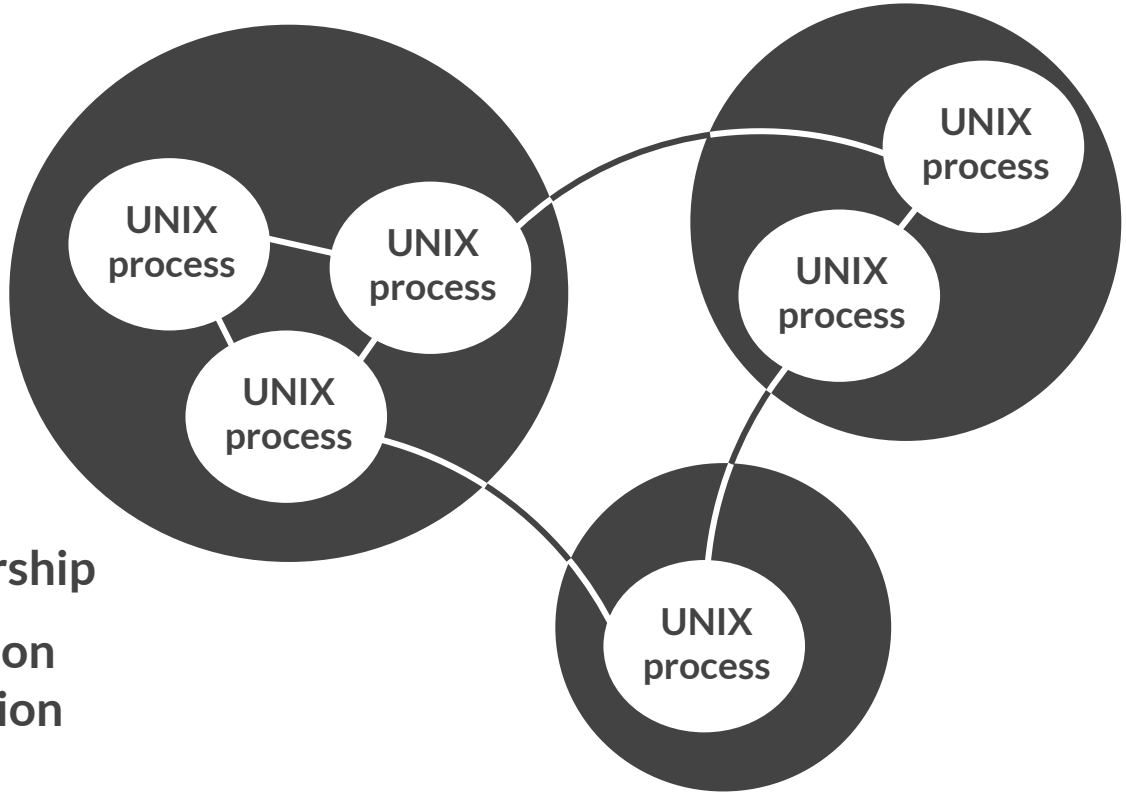


GopherCon 2014
Denver, CO

Petar Maymounkov
XData Open-Source Initiative
Data Tactics Corp

Distributed Control

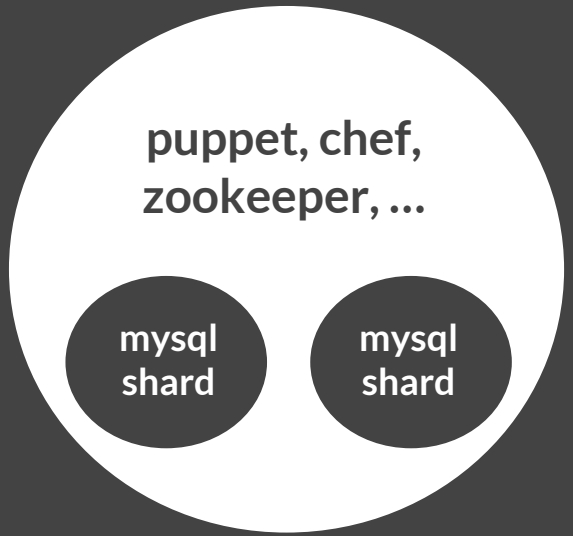
The ability of one APP process to affect and be affected by its peers.



Host Membership

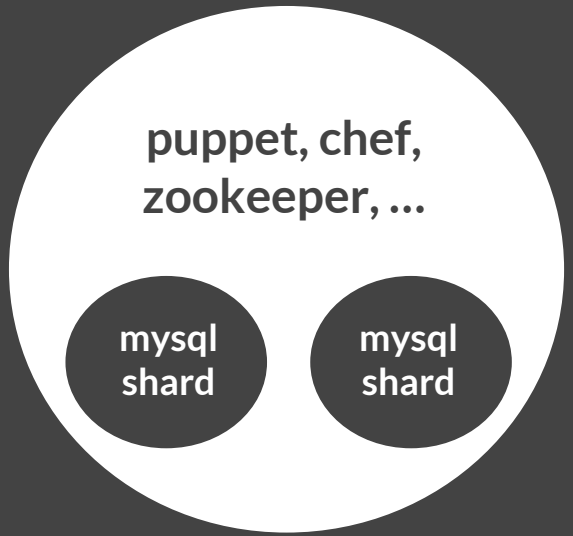
Communication
Synchronization
Process

Distributed Control Outside



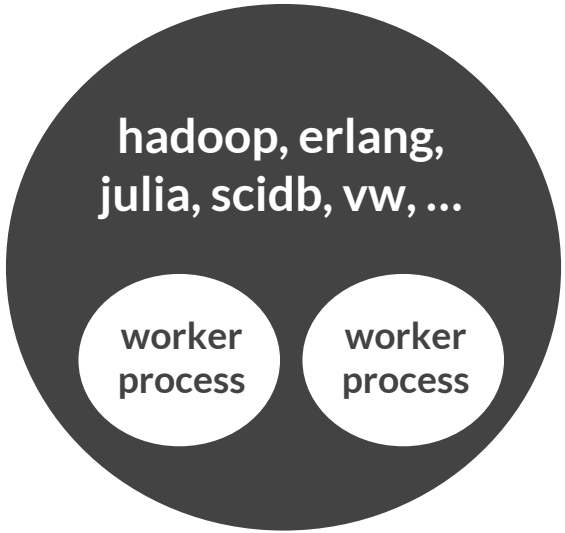
Complex config
Complex maintainer manual

Distributed Control Outside



Complex config
Complex maintainer manual

Distributed Control Inside

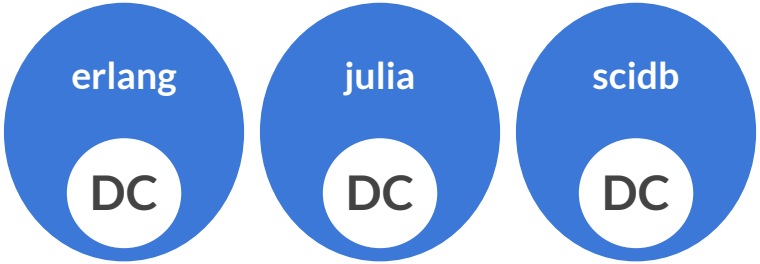


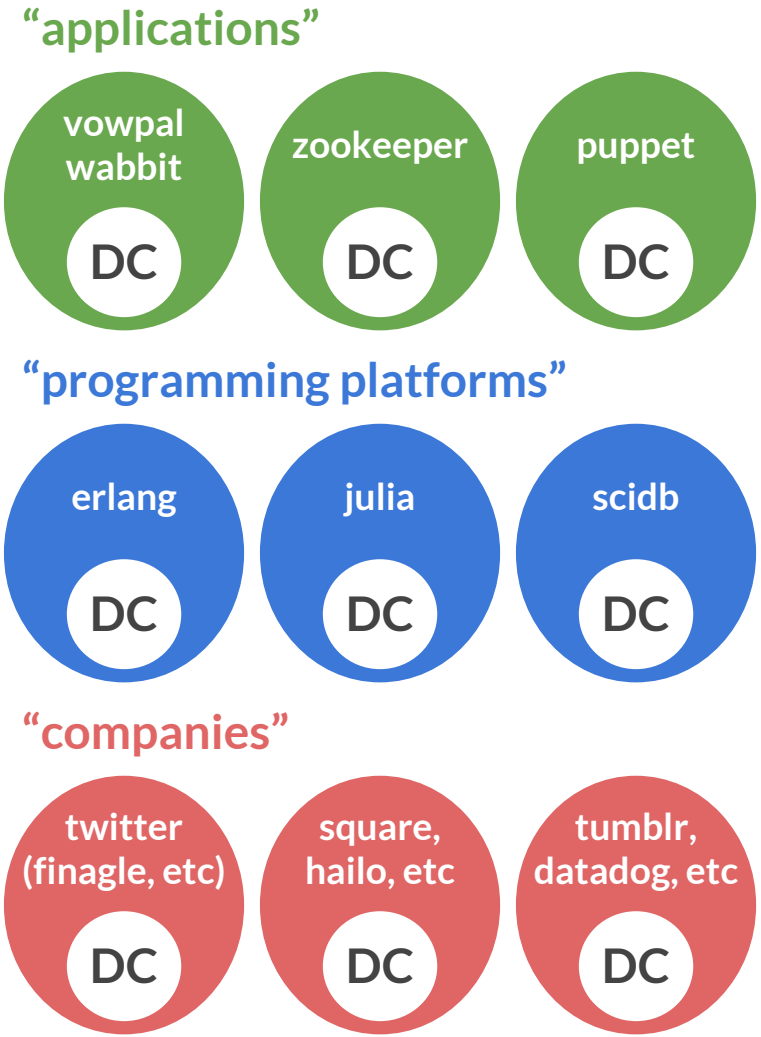
Simple config
Hardly any maintainer manual, but ...

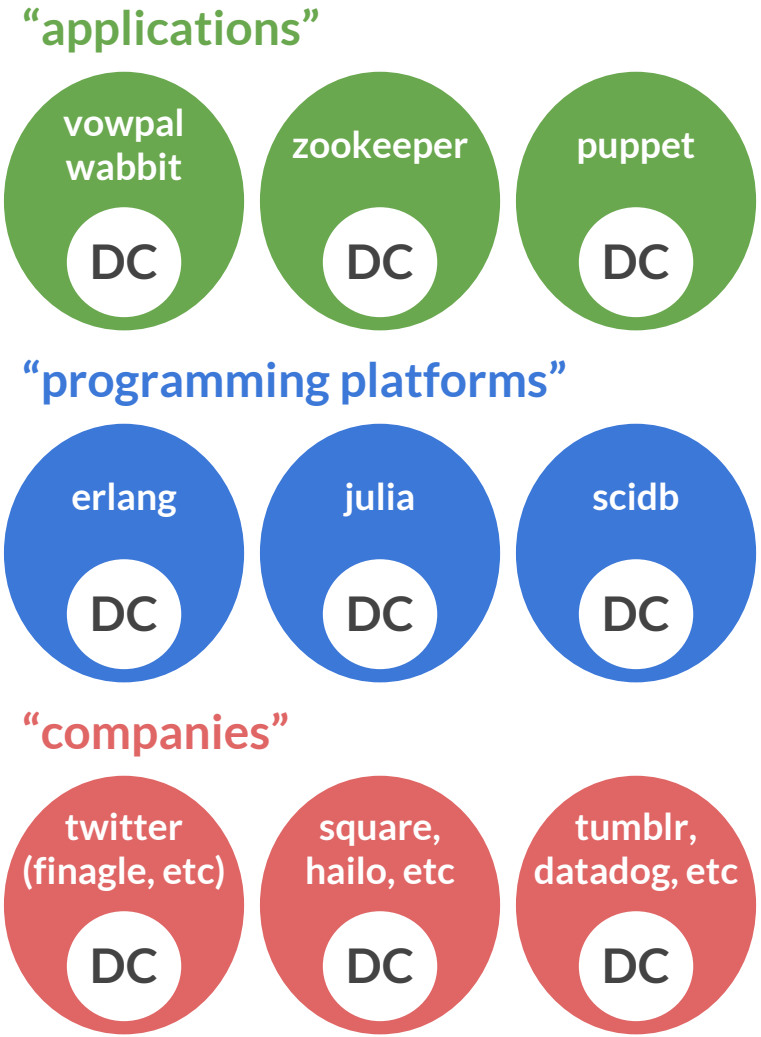
“applications”

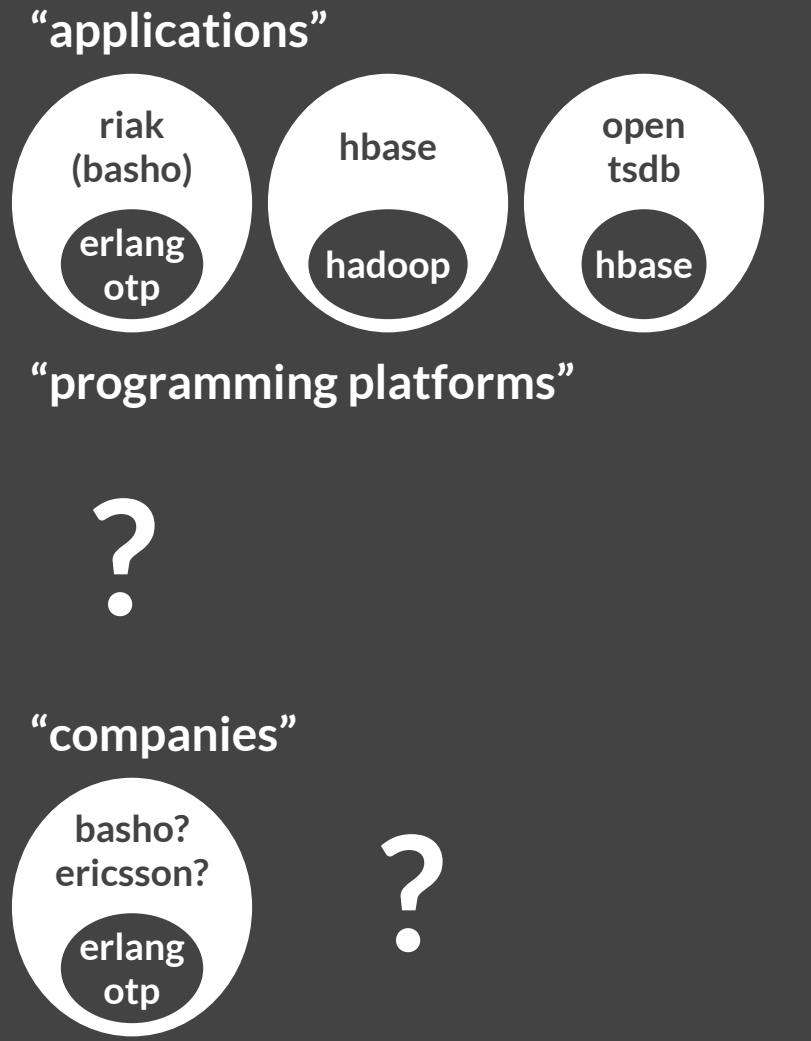
“applications”

“programming platforms”




“applications”
vowpal wabbit
DC
zookeeper
DC
puppet
DC
“programming platforms”
erlang
DC
julia
DC
scidb
DC
“companies”
twitter (finagle, etc)
DC
square, hailo, etc
DC
tumblr, datadog, etc
DC


"applications"
vowpal wabbit
DC
zookeeper
DC
puppet
DC
"programming platforms"
erlang
DC
julia
DC
scidb
DC
"companies"
twitter (finagle, etc)
DC
square, hailo, etc
DC
tumblr, datadog, etc
DC

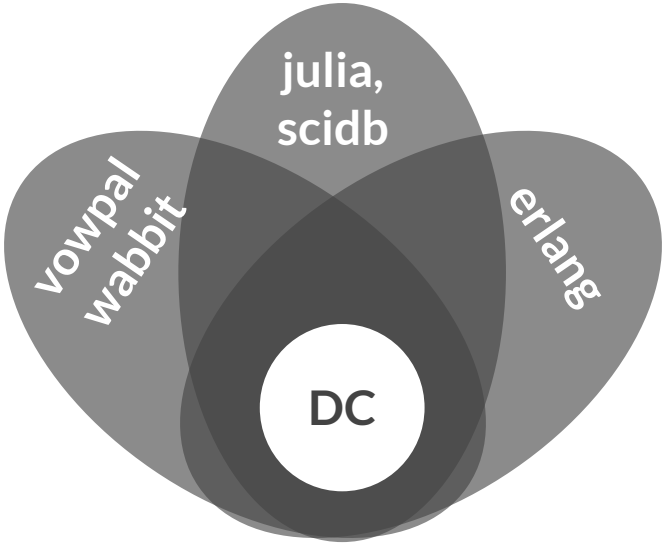

"applications"
riak (basho)
erlang otp
hbase
hadoop
open tsdb
hbase
"programming platforms"
?
"companies"
basho? ericsson?
erlang otp
?

To benefit more causes, one needs ...

Language-agnostic
interoperability

while also

Language-like
easy to think with



Choosing an API

Semantics and Syntax

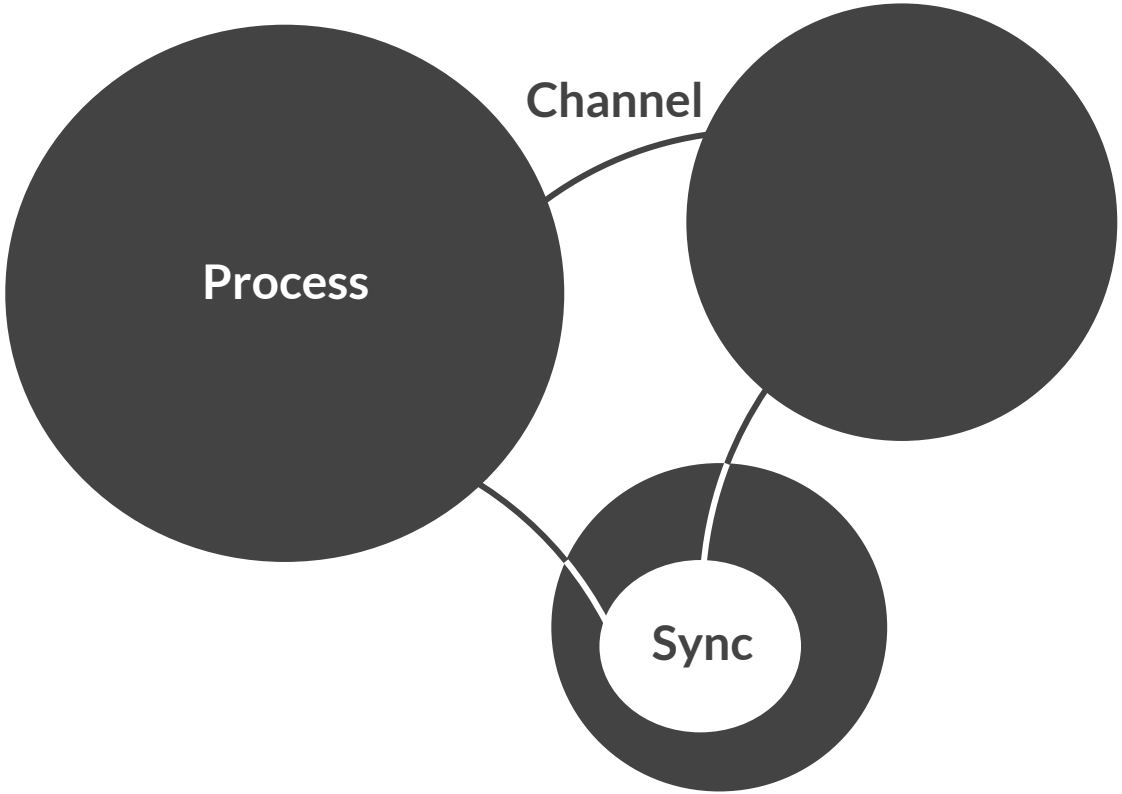
API: Semantics

CSP + host membership oracle

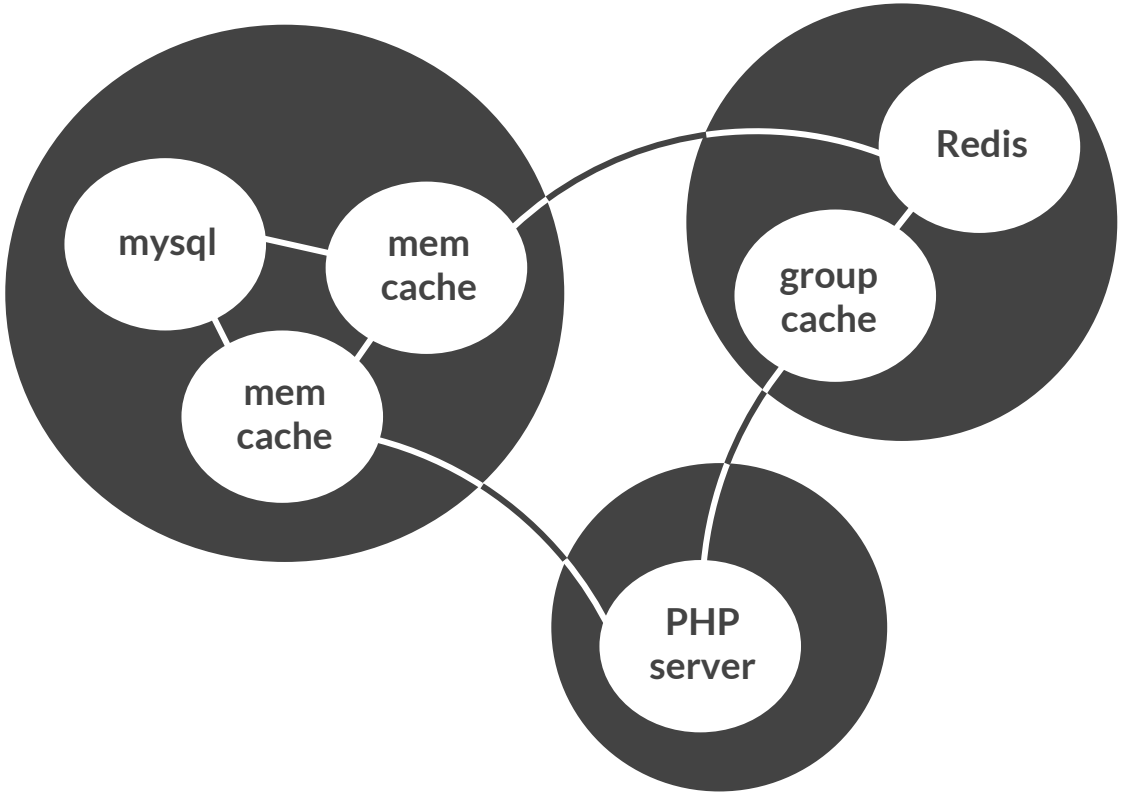
Membership

```
type Membership interface {
    WaitForJoinOrLeave() Event
}

type WorkerID string

type Joined WorkerID // Event
type Left   WorkerID // Event
```

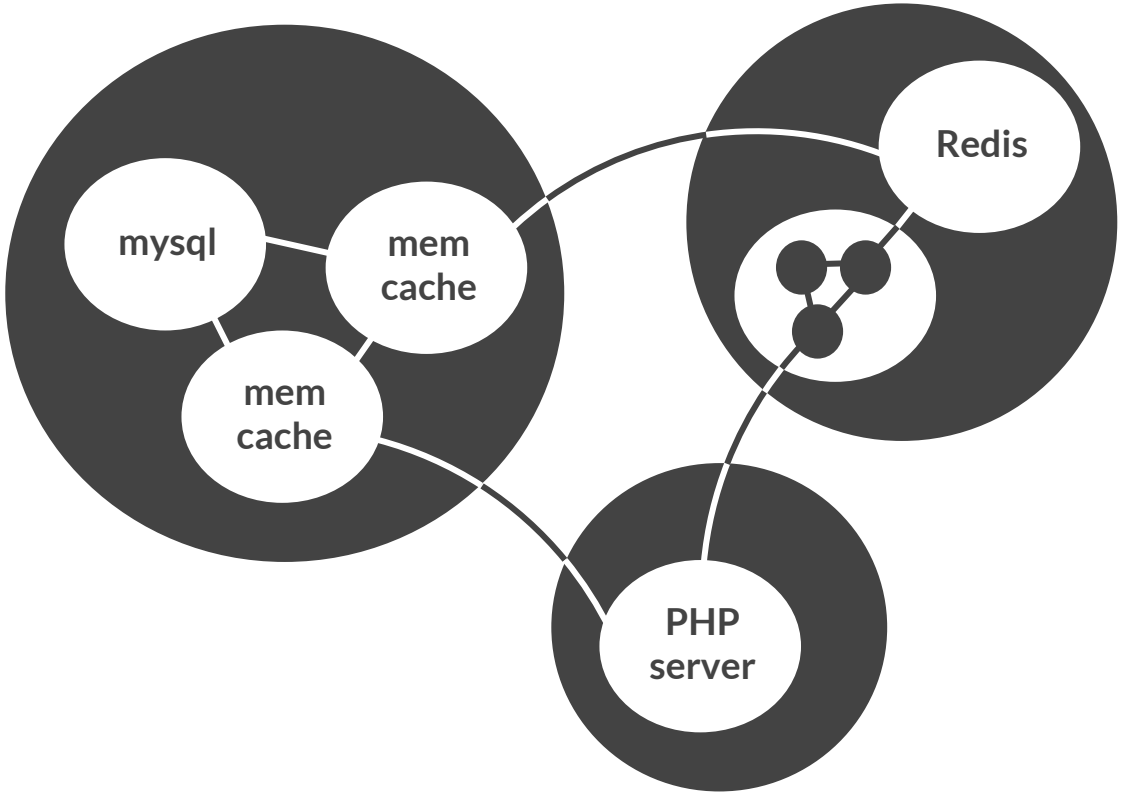
Hoare's CSP



Cluster-level design



Cluster- and process-level design



Syntax of native CSP in Go

```
ch := make(chan interface{})
ch <- 1
<-ch
close(ch)
```

Communication

```
select {
case <-ch:
case ch <- s:
default:
}
```

Synchronization

```
go func() { … }
```

Process

Rewrite in package-style

```
func MakeChan(cap int) Chan
type Chan interface {
     Send(interface{})
     Recv() (interface{}, bool)
     Close()
}
```

Communication

```
type Clause struct
{
     Send Chan
     Recv Chan
     Exit Proc
}
```

```
func MakeSel() Sel
type Sel interface {
     Start([]Clause)
     Wait() int
}
```

Synchronization

```
func MakeProc()Proc
type Proc interface {
     Start(func())
     Wait()
}
```

Process

Introduce location argument (=GoCircuit)

```
func MakeChan(where WorkerID, cap int) Chan
type Chan interface {
     Send(interface{})
     Recv() (interface{}, bool)
     Close()
}
```

Communication

```
type Clause struct
{
     Send Chan
     Recv Chan
     Exit Proc
}
```

```
func MakeSel(where WorkerID) Sel
type Sel interface {
     Start([]Clause)
     Wait() int
}
```

Synchronization

```
func MakeProc(where WorkerID)Proc
type Proc interface {
     Start(func())
     Wait()
}
```

Process

Replace goroutines with UNIX processes

```
func MakeChan(where WorkerID, cap int) Chan
type Chan interface {
     Send([]byte)
     Recv() ([]byte, bool)
     Close()
}
```

Communication

```
type Clause struct
{
     Send Chan
     Recv Chan
     Exit Proc
}
```

```
func MakeSel(where WorkerID) Sel
type Sel interface {
     Start([]Clause)
     Wait() int
}
```

Synchronization

```
type Command struct {
     Env  []string
     Path string
     Args []string
}
```

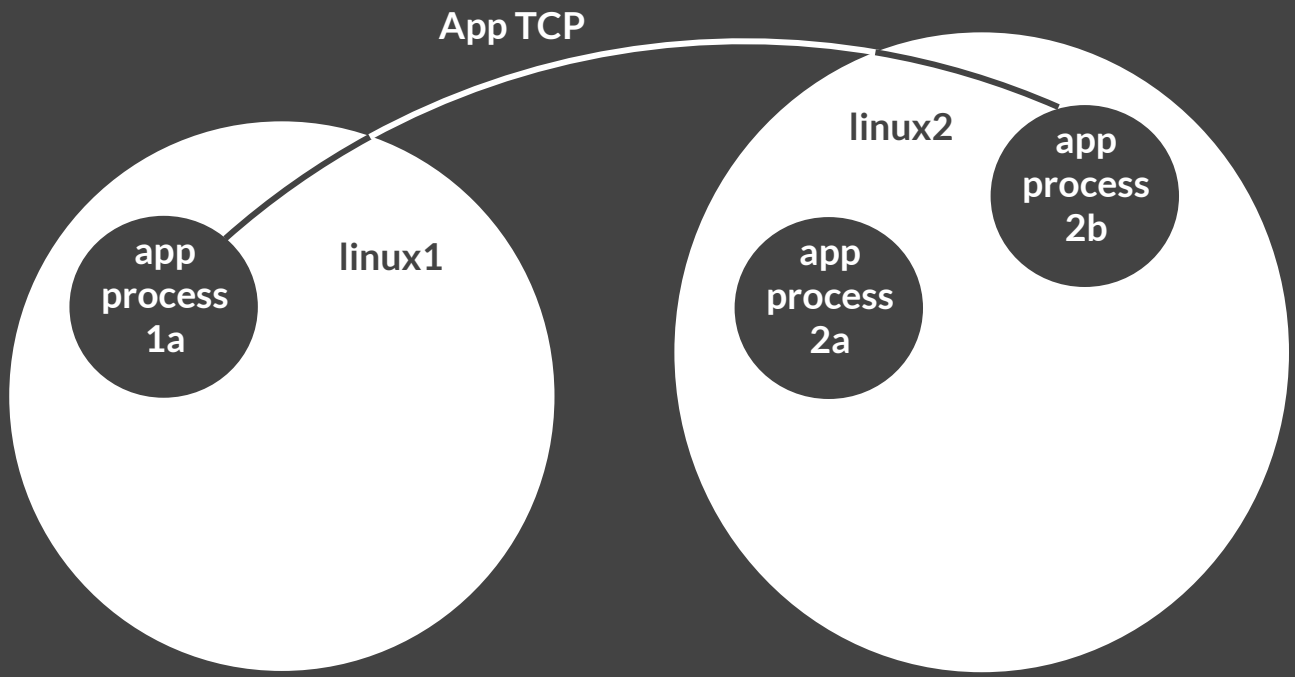
```
func MakeProc(where WorkerID)Proc
type Proc interface {
     Start(cmd Command)
     Wait()
}
```

Process

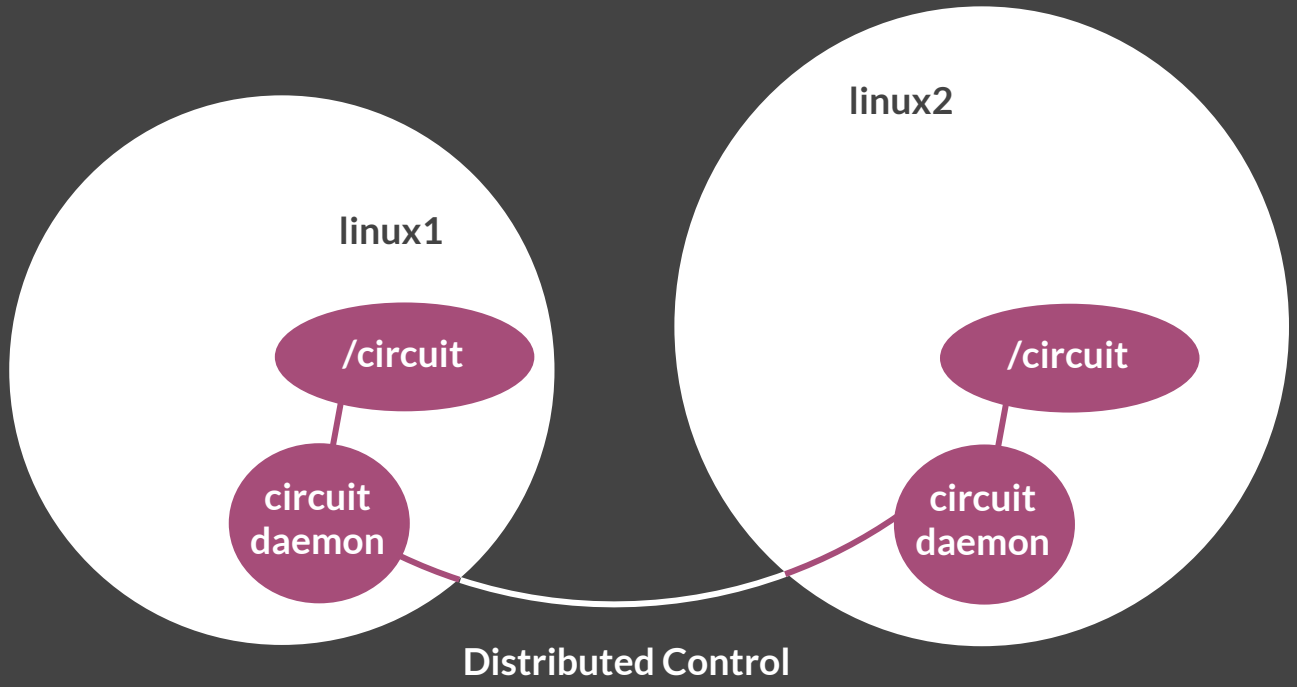
API: Syntax

Embed the API into the local file system

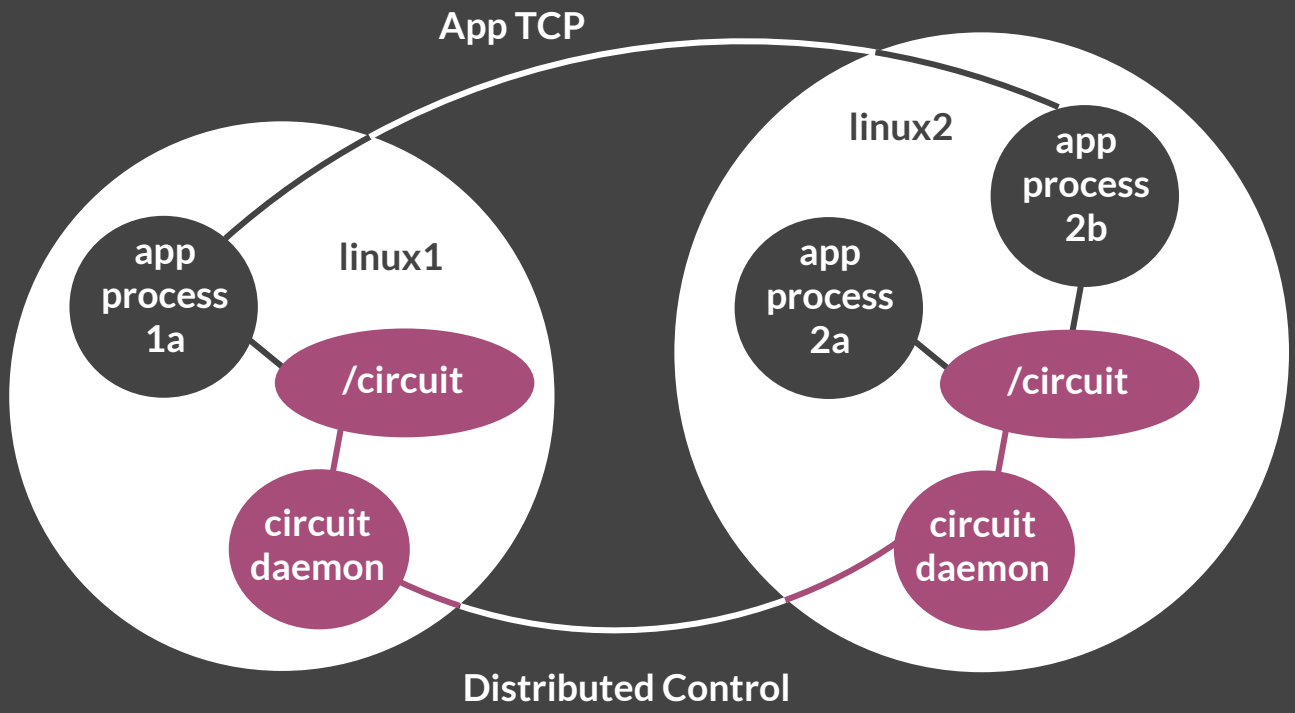
App and circuit relationship 1/3



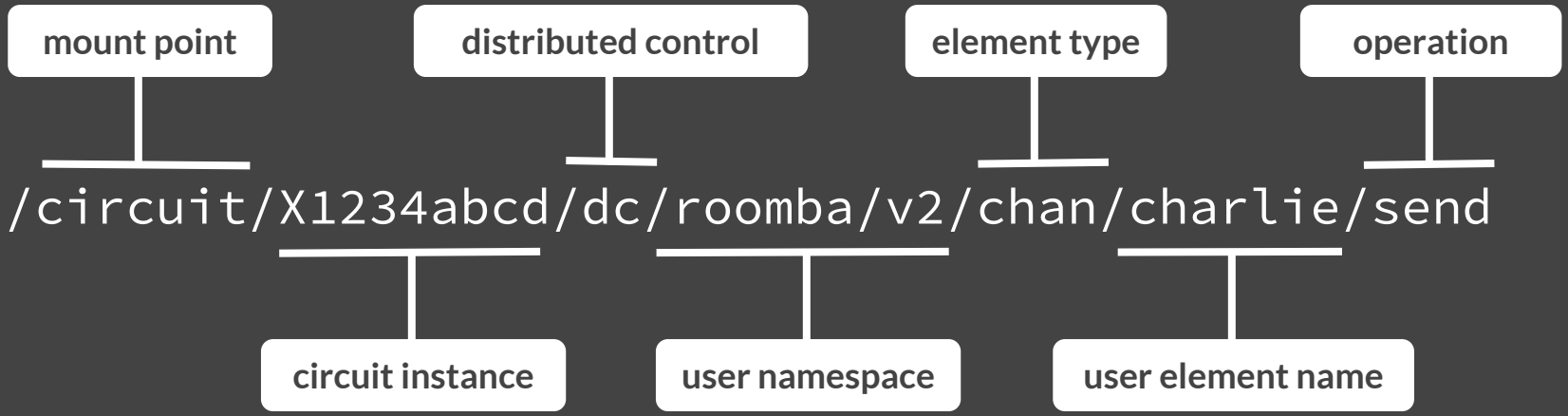
App and circuit relationship 2/3



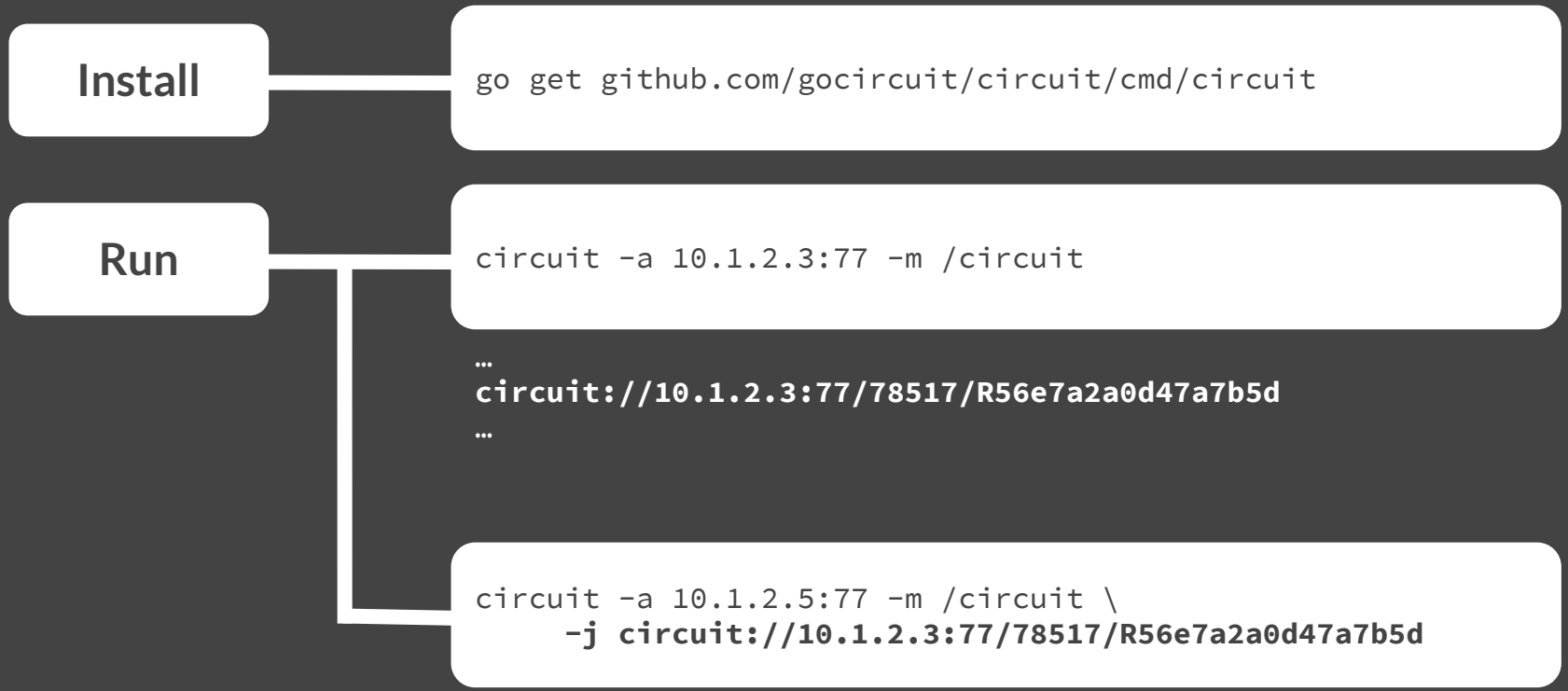
App and circuit relationship 3/3



File-system CSP interface for Distributed Control



Install. Run. Use.


Install
go get github.com/gocircuit/circuit/cmd/circuit
Run
circuit -a 10.1.2.3:77 -m /circuit
…
circuit://10.1.2.3:77/78517/R56e7a2a0d47a7b5d
…
circuit -a 10.1.2.5:77 -m /circuit \
-j circuit://10.1.2.3:77/78517/R56e7a2a0d47a7b5d

Membership

```
cd /circuit
ls
```

```
dr-xr-xr-x  1 petar  staff  3 Apr 16 09:50 X13c6fa9d732d5d38/
dr-xr-xr-x  1 petar  staff  3 Apr 16 07:10 X93bcde820efa3567/
```

```
cd X93bcde820efa3567
ls
```

```
drwxrwxrwx  1 petar  staff     4 Apr 16 09:53 dc/
drwxr-xr-x  1 petar  staff  1292 Apr  9 07:38 fs/
-r--r--r--  1 petar  staff     0 Apr 16 09:53 help
dr-xr-xr-x  1 petar  staff     5 Apr 16 09:53 sys/
```

```
cd dc
```

/circuit/X93bcde820efa3567/dc**/proc/pippi/**

Process

```
cd proc
mkdir pippi
ls
```

```
-r--r--r-- env
-r--r--r-- error
-r--r--r-- help
-rw-rw-rw- run
--w--w--w- signal
-r--r--r-- stat
-r--r--r-- stderr
--w--w--w- stdin
-r--r--r-- stdout
-r--r--r-- waitexit
```

```
echo << EOF >> run
{
     “env”:  [“TERM=circuit”],
     “path”: “/bin/ls”,
     “args”: [“-a”, “/”]
}
EOF
```

```
cat waitexit
```

/circuit/X93bcde820efa3567/dc**/proc/pippi/**

Process

```
cd proc
mkdir pippi
ls
```

```
-r--r--r-- env
-r--r--r-- error
-r--r--r-- help
-rw-rw-rw- run
--w--w--w- signal
-r--r--r-- stat
-r--r--r-- stderr
--w--w--w- stdin
-r--r--r-- stdout
-r--r--r-- waitexit
```

```
echo << EOF >> run
{
     “env”:  [“TERM=circuit”],
     “path”: “/bin/ls”,
     “args”: [“-a”, “/”]
}
EOF
```

```
yes >> stdin &
```

```
cat stdout >> /dev/null &
```

```
cat waitexit
```

/circuit/X93bcde820efa3567/dc/**chan/charlie/**

Channel

```
cd chan
mkdir charlie
ls
```

```
-rw-rw-rw- cap
--w--w--w- close
-r--r--r-- error
-r--r--r-- help
-r--r--r-- recv
--w--w--w- send
-r--r--r-- stat
-r--r--r-- tryrecv
--w--w--w- trysend
```

```
echo “0” >> cap
echo “¡Hello, world!” >> send
```

```
cat recv
```

```
¡Hello, world!
```

```
echo “close” >> close
```

/circuit/X93bcde820efa3567/dc/**select/sarah/**

Select

```
cd select
mkdir sarah
ls
```

```
--w--w--w- abort
-r--r--r-- error
-r--r--r-- help
-rw-rw-rw- select
-r--r--r-- wait
```

```
echo << EOF >> select
[
      {“op”: “r”, “file”: “…/proc/p0/waitexit”},
      {“op”: “w”, “file”: “…/chan/c1/send”},
      {“op”: “r”, “file”: “…/chan/c2/recv”}
]
EOF
```

```
cat wait
```

```
{
      “clause”: 2,
      “file”:   “read.2”
}
```

/circuit/X93bcde820efa3567/dc/**select/sarah/**

Select

```
cd select
mkdir sarah
ls
```

```
--w--w--w- abort
-r--r--r-- error
-r--r--r-- help
-rw-rw-rw- select
-r--r--r-- wait
```

```
echo << EOF >> select
[
      {“op”: “r”, “file”: “…/proc/p0/waitexit”},
      {“op”: “w”, “file”: “…/chan/c1/waitsend”},
      {“op”: “r”, “file”: “…/chan/c2/waitrecv”}
]
EOF
```

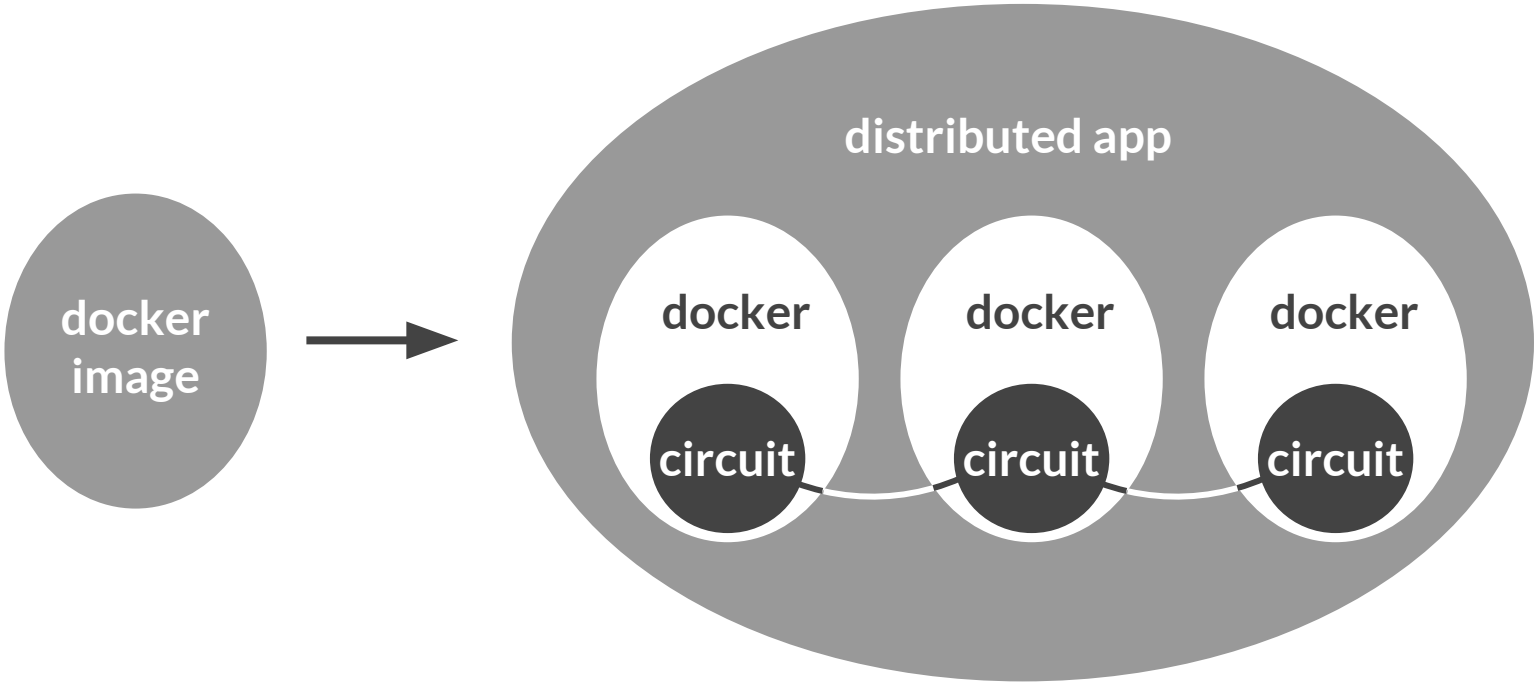
```
echo “hola” >> …/c2/send
```

```
cat wait
```

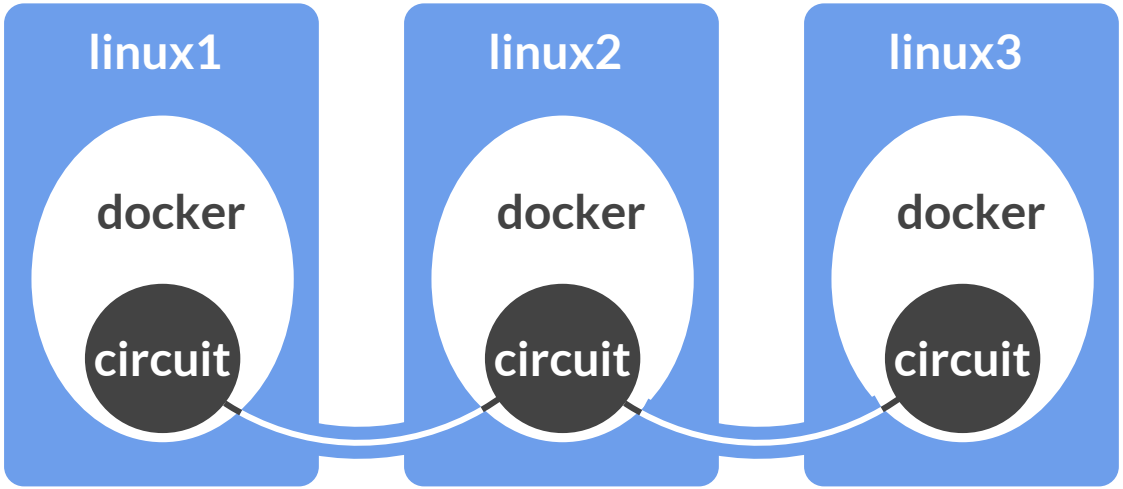
```
{
      “clause”: 2,
      “file”:   “read.2”
}
```

Applications

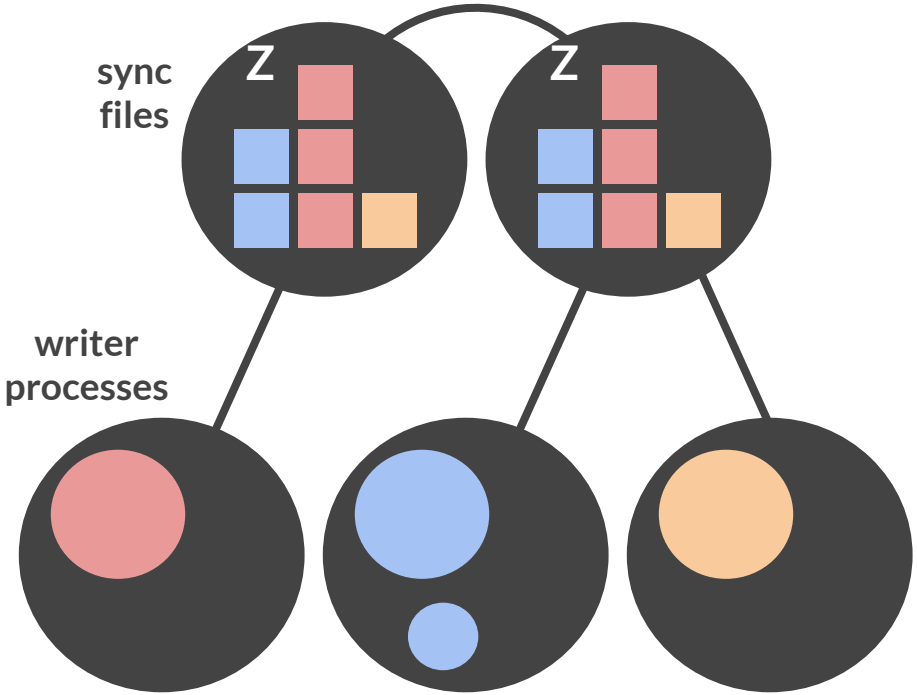
Distributed Binaries



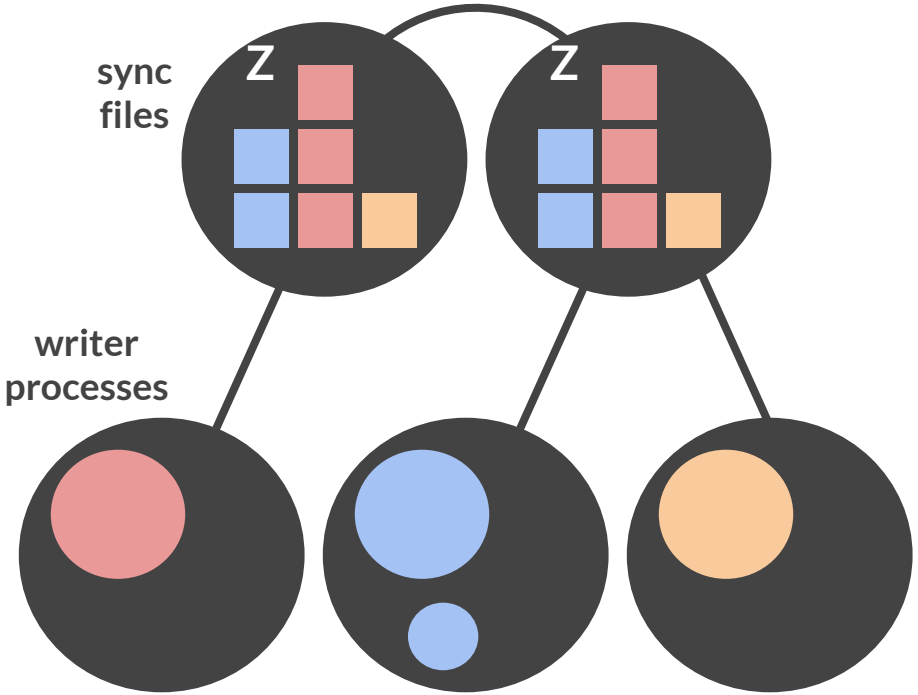
Security

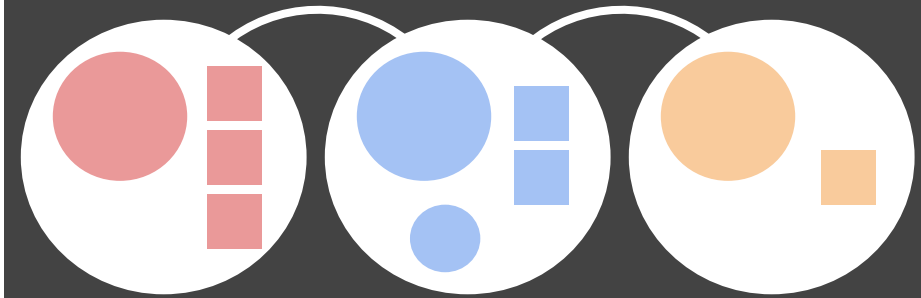

linux1
docker
circuit
linux2
docker
circuit
linux3
docker
circuit

Zookeeper, Chubby, etc.

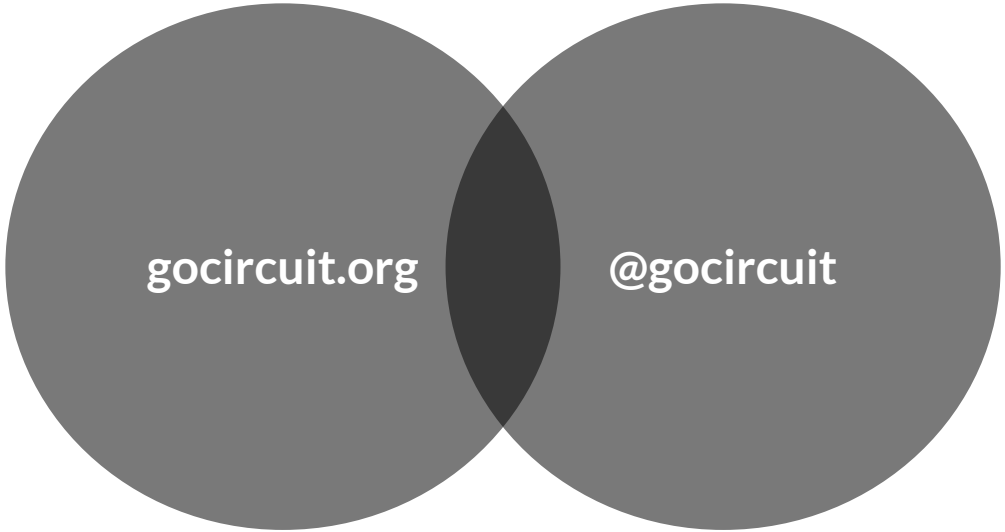


Zookeeper, Chubby, etc. possibly obsolete





Thank you



circuit = gocircuit + file system interface

gocircuit: heavily used, bugless over 1 year

file system interface: about 3 months old, independent, simple code