[544] OpenLambda Architecture

Tyler Caraza-Harter

Event

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

Event

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

SB

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

OpenLambda uses a custom container implementation (SOCK) instead of Docker optimized for Lambdas, but which could be used for other systems.

mem pool

mem

ing)

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

Event

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

SB

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

- cgroups provide performance isolation
- namespaces provide logical isolation
- chroot provides logical isolation for FS
- seccomp blocks "dangerous" syscalls

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

Event

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

SB

SB

SB

SB

SB

sandboxes

a new sandbox can be created as a fork of a parent sandbox

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

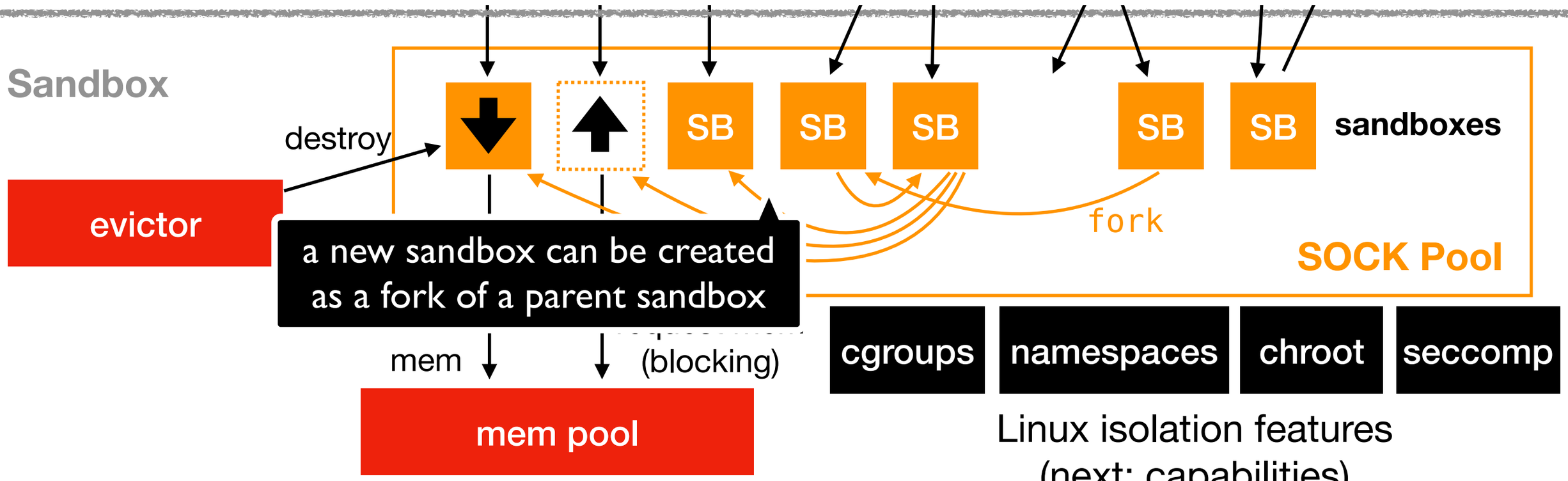
```
type SandboxPool interface {
  Create(parent Sandbox, isLeaf bool, codeDir, ...) (Sandbox, error)
  ...
}
```

where to fork from (may be null)

is this a handler (true) or a Zygote (false)

path to Lambda code

Sandbox API



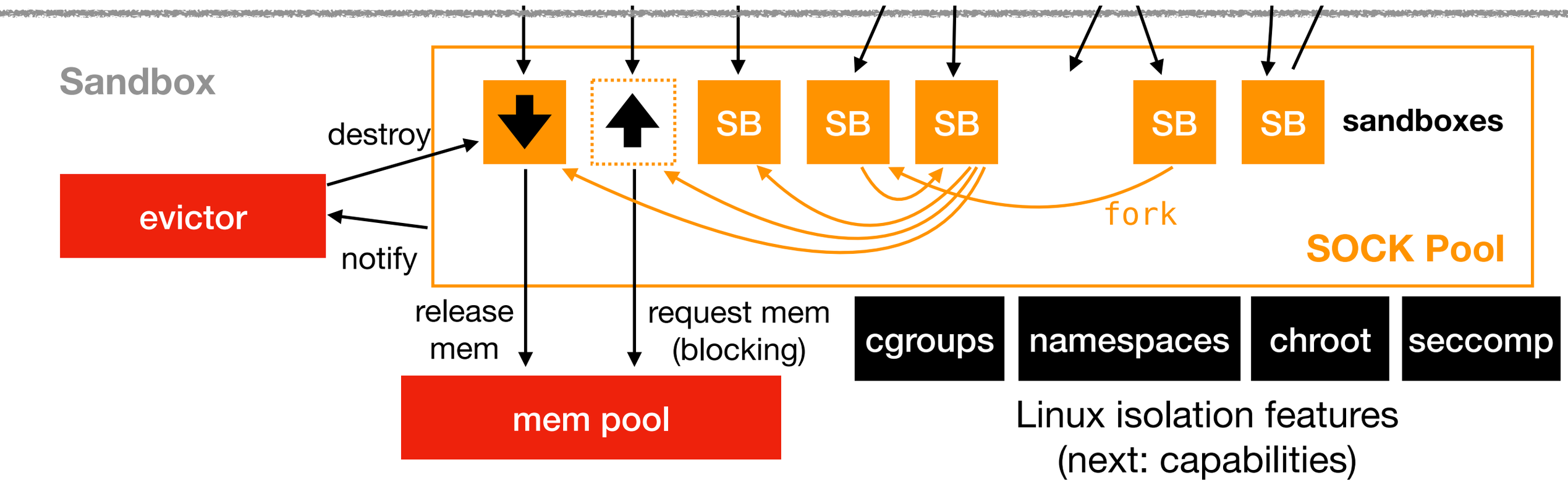
```
type SandboxPool interface {
  Create(parent Sandbox, isLeaf bool, codeDir, ...) (Sandbox, error)
  ...
}
```

where to fork from (may be null)

is this a handler (true) or a Zygote (false)

path to Lambda code

```
type Sandbox interface {
  Client() (*http.Client)
  Pause() error
  Unpause() error
  ...
  Destroy(reason string)
  DestroyIfPaused(reason string)
}
```



```
type SandboxPool interface {
  Create(parent Sandbox, isLeaf bool, codeDir, ...) (Sandbox, error)
  ...
}
```

where to fork from (may be null)

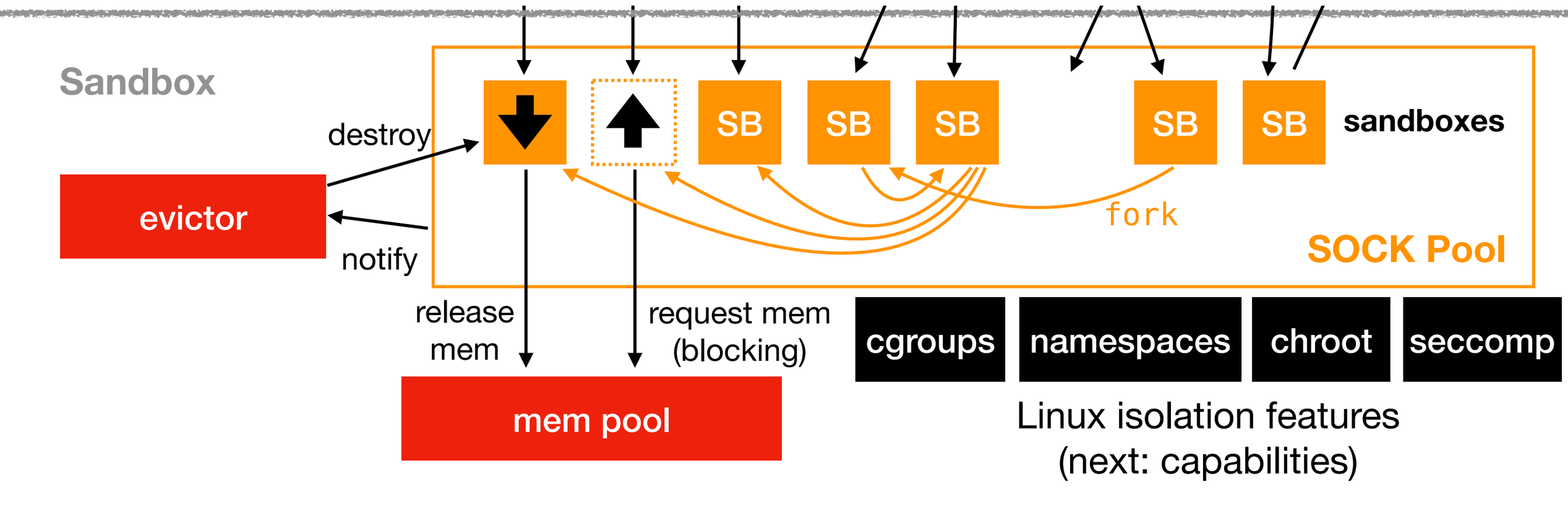
is this a handler (true) or a Zygote (false)

path to Lambda code

```
type Sandbox interface {
  Client() (*http.Client)
  Pause() error
  Unpause() error
  ...
  Destroy(reason string)
  DestroyIfPaused(reason string)
}
```

forward web requests

are Lambda processes schedulable by Linux?

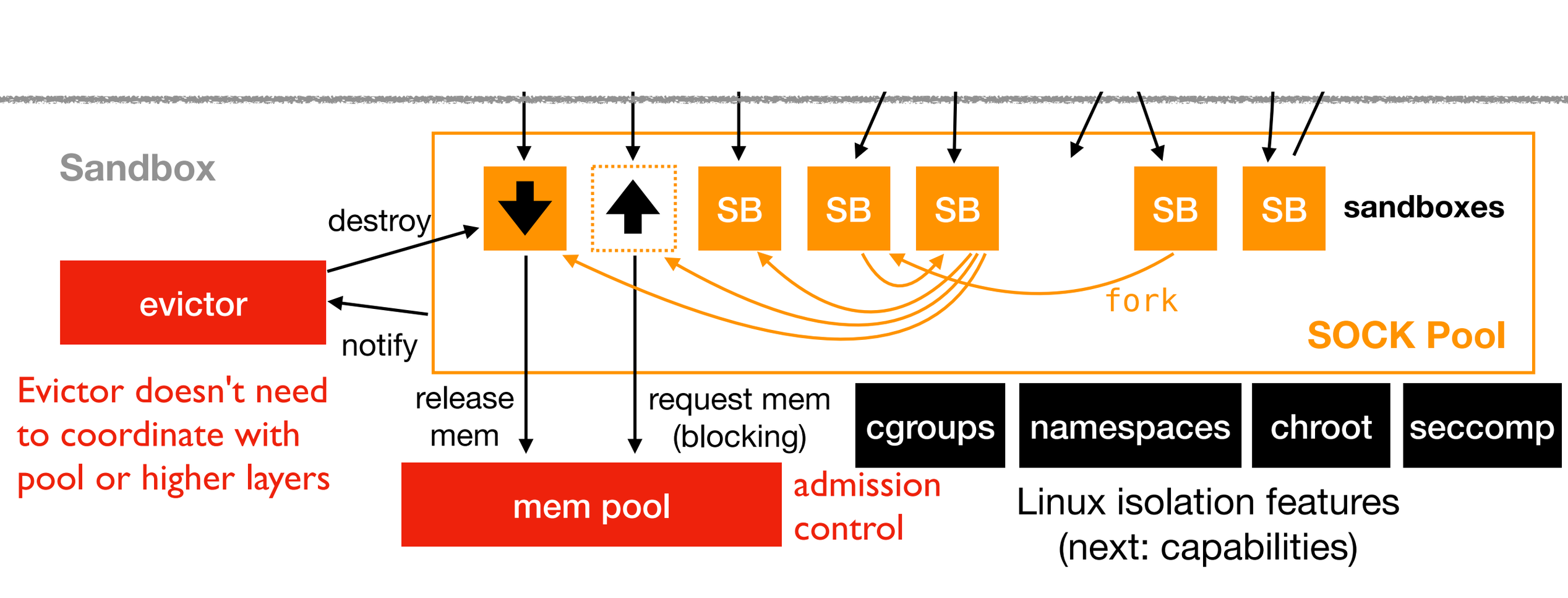


```
type safeSandbox struct {
        Sandbox

        sync.Mutex
        paused          bool
        dead            error
        eventHandlers []SandboxEventFunc
}
```

safeSandbox is a Sandbox and wraps every other Sandbox implementation

- Sandbox API is threadsafe
- If any call returns any error, Sandbox is automatically destroyed (caller can just stop using it)
- Any call on destroyed Sandbox returns error, causes no harm
- notifies listeners (particularly evictor) that can operate independently



Event

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

Event

invoke

Lambda

queue

code puller

code store

package puller

PyPI repo

Handles events; responsibilities:
1. fetch handler code (check for updates)
2. fetch package dependencies
3. allocate Sandboxes to run handlers

λ instances C1 D1 D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

One sandbox pool used by three subsystems: (1) Lambda instances, (2) import cache, (3) package puller. Previous instances and cache used different pools, and package puller was (unsafely) not sandboxed.

dest

evictor

noti

SB SB sandboxes

ork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups namespaces chroot seccomp

Linux isolation features
(next: capabilities)

Event

OpenLambda now uses a generic HTTP server as code store. Checks for new versions every **N** seconds (configurable) with `If-Modified-Since` header

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups

namespaces

chroot

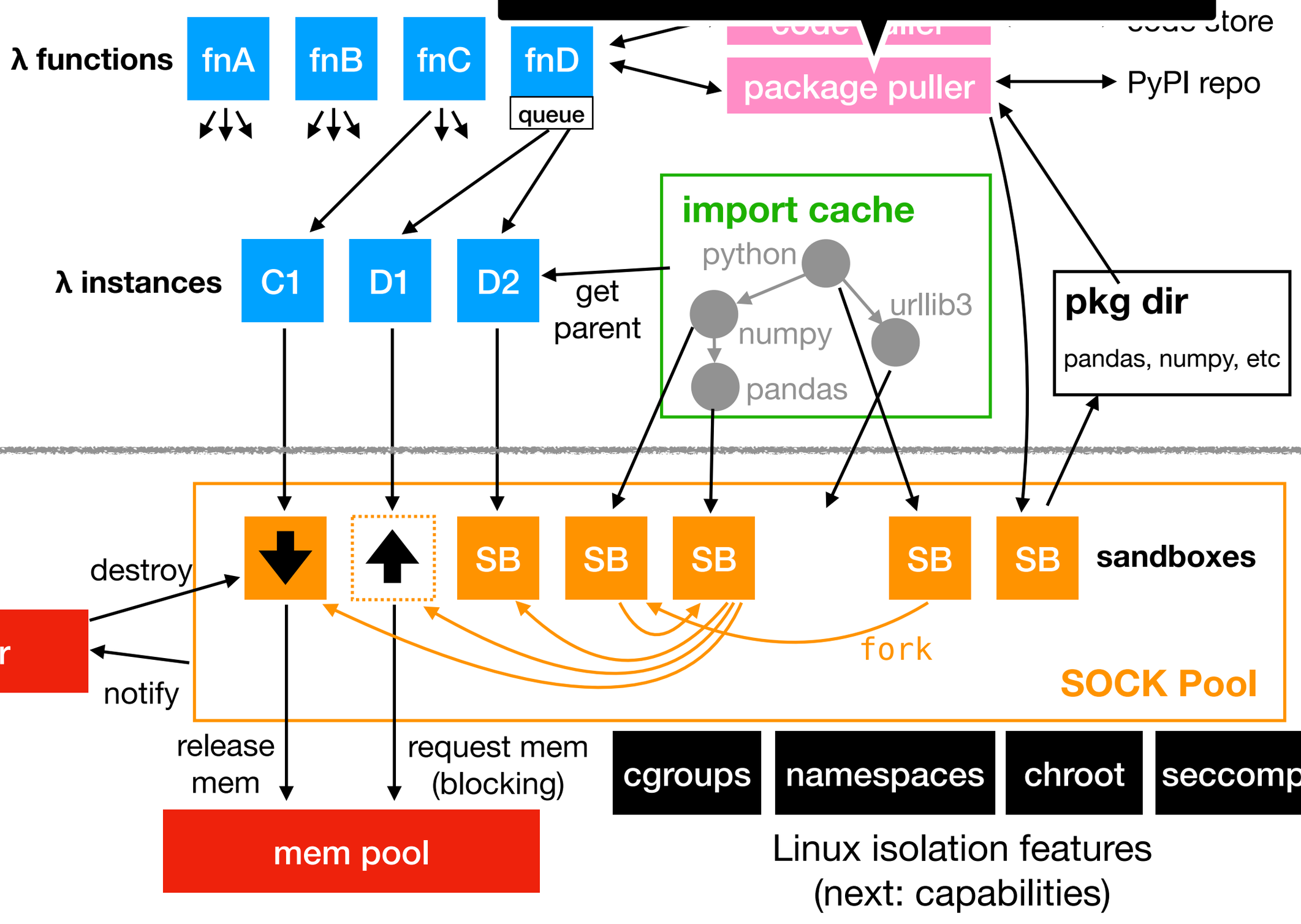
seccomp

Linux isolation features
(next: capabilities)

Event

package puller uses Sandoxes to install packages locally because pip install may execute malicious code. OpenLambda now recursively discovers dependencies and support multiple versions of the same pkg.

Lambda



Event

Lambda

invoke

- pulls code, packages
- kill stale instances if code changes
- decides number of instances based on queue depth, avg processing time
- returns **`HTTP 429 Too Many`** if queue is full

λ functions

fnA fnB fnC fnD

queue

queue

import cache

λ instances

C1 D1 D2

Basically a robust, virtual Sandbox
- *sometimes* backed by physical SB
- masks Sandbox failure by restarting them

Sandbox

destroy

evictor

notify

SB SB SB SB SB SB

sandboxes

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

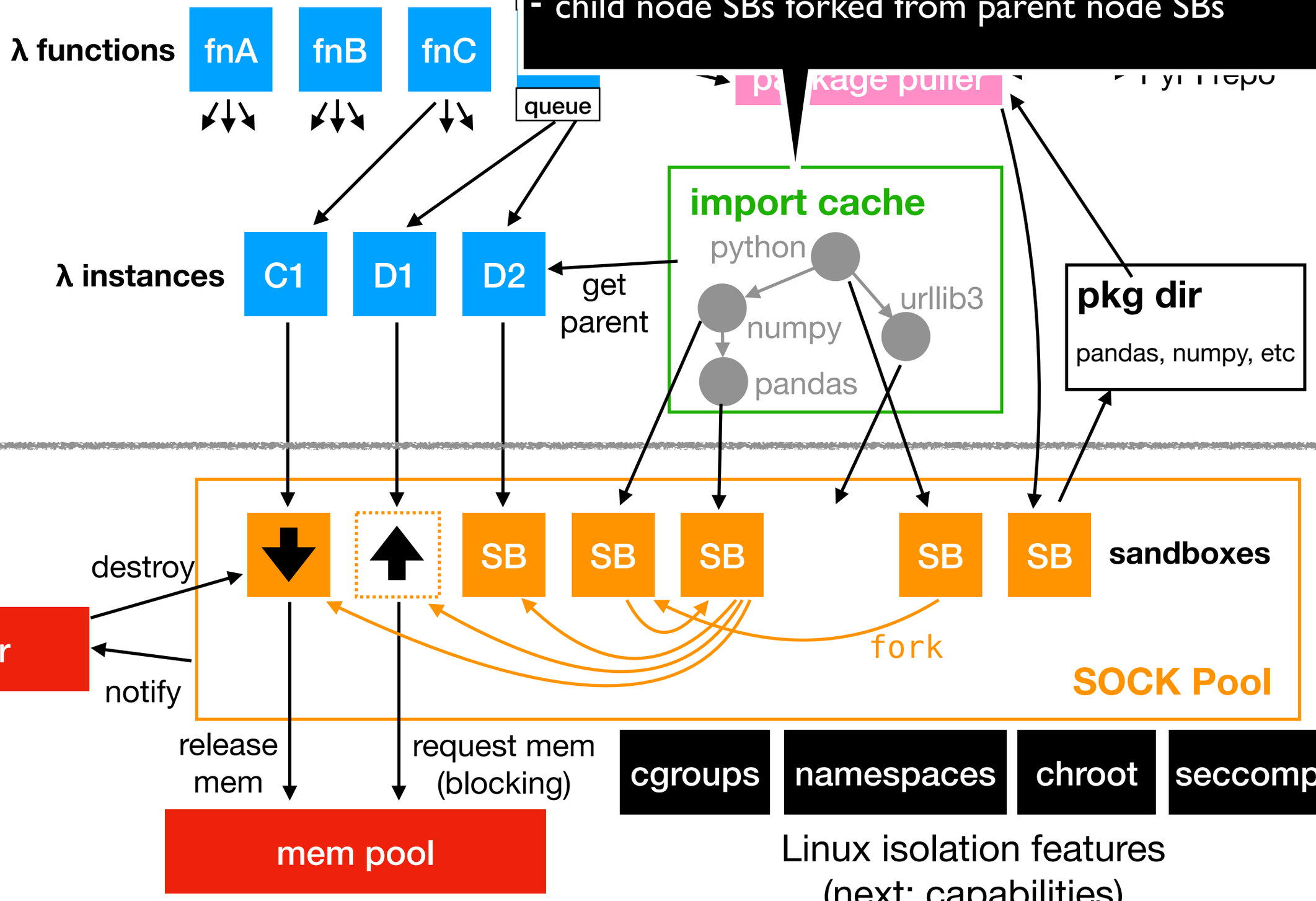
cgroups namespaces chroot seccomp

Linux isolation features
(next: capabilities)

Event

Lambda instances request parent zygotes from import cache to fork instance sandboxes
- zygote tree specified at startup
- each node not initially backed by actual Sandbox
- zygote created/evicted based on workload
- child node SBs forked from parent node SBs

Lambda



Event

could be things like cron event, DB update, message, etc. OL currently only supports web req events

invoke

Lambda

λ functions

fnA

fnB

fnC

queue

fnD

queue

code puller

code store

package puller

PyPI repo

import cache

python

urllib3

numpy

pandas

pkg dir

pandas, numpy, etc

λ instances

C1

D1

D2

get parent

Sandbox

destroy

evictor

notify

SB

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

release mem

request mem (blocking)

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)