

OpenLambda Architecture

Tyler Caraza-Harter

Event

invoke

Lambda

λ functions

fnA

fnB

fnC

fnD

queue

queue

code puller

- code store

package puller

- PyPI repo

λ instances

11/11/2019

100

100

get
parent

```
import cache
```

python



urllib3

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

release
mem

request mem
(blocking)

mem pool

SB

SB

SB

SE

SB

sandboxes

fork

SOCK Pool



namespaces

chroot

seccomp

Linux isolation features (next: capabilities)

Event

Lambda

λ functions

fnA

fnB

fnC

fnD

queue

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

numpy

pandas

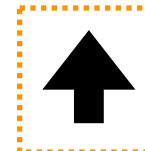
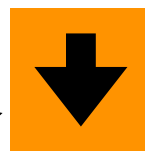
urllib3

pkg dir

pandas, numpy, etc

Sandbox

destroy



SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

mem
(ing)

cgroups

namespaces

chroot

seccomp

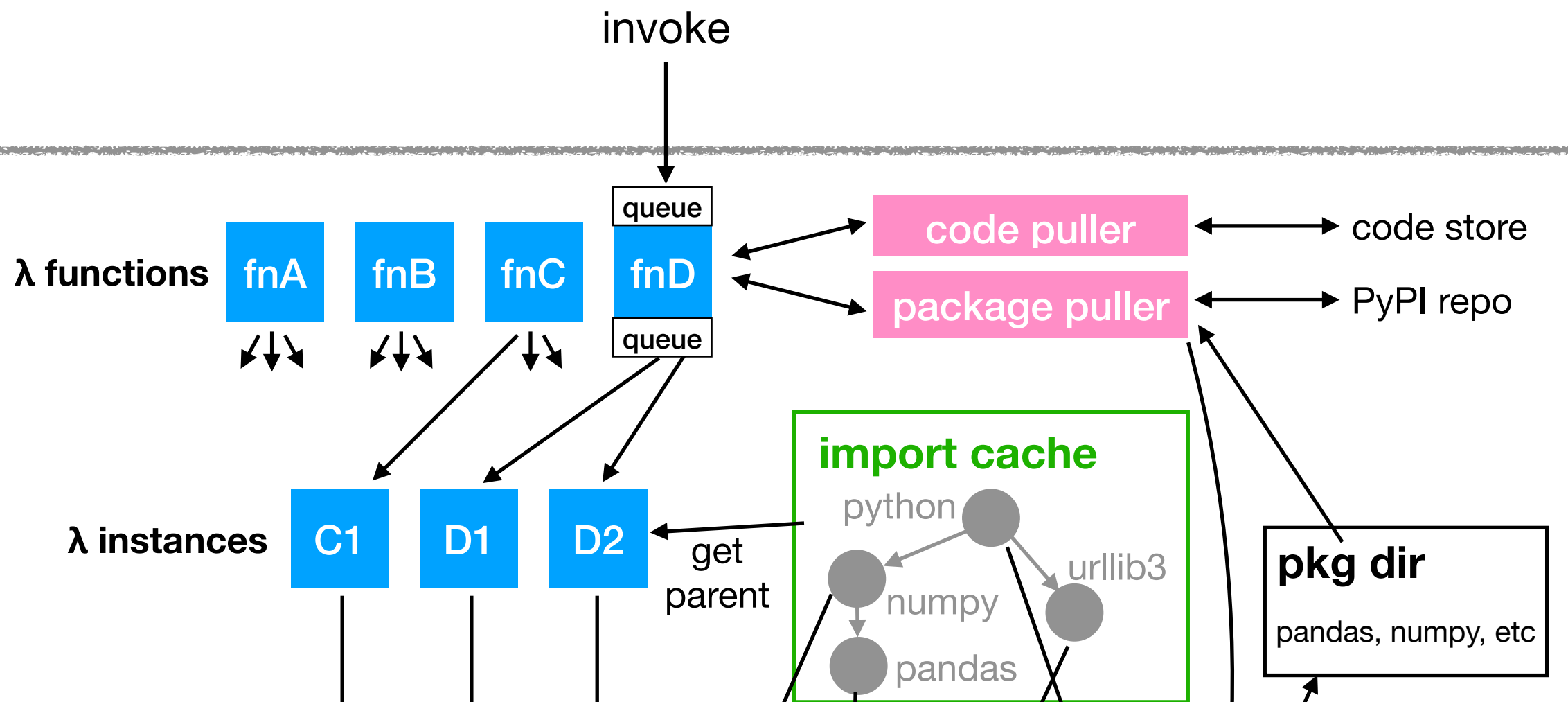
Linux isolation features
(next: capabilities)

mem pool

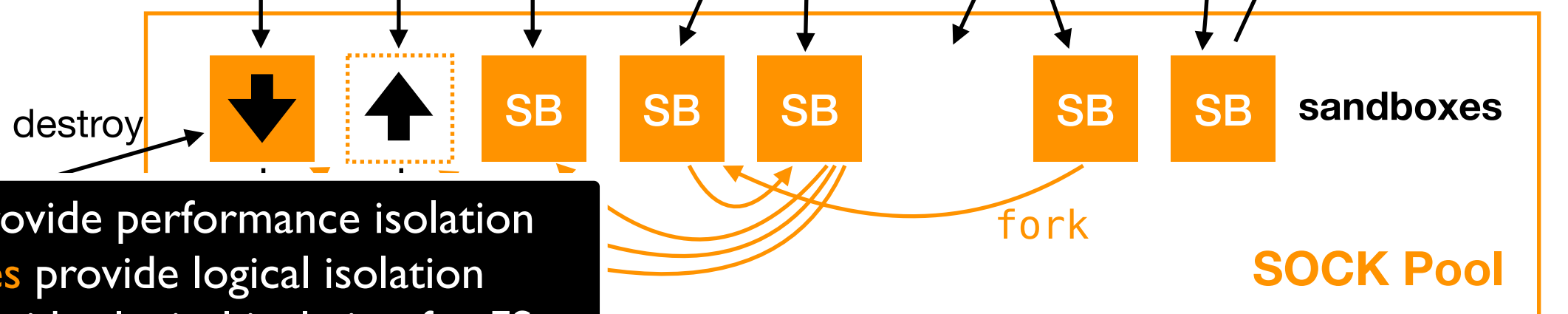
OpenLambda uses a custom container implementation (SOCK) instead of Docker optimized for Lambdas, but which could be used for other systems.

Event

Lambda



Sandbox



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

mem pool

cgroups

namespaces

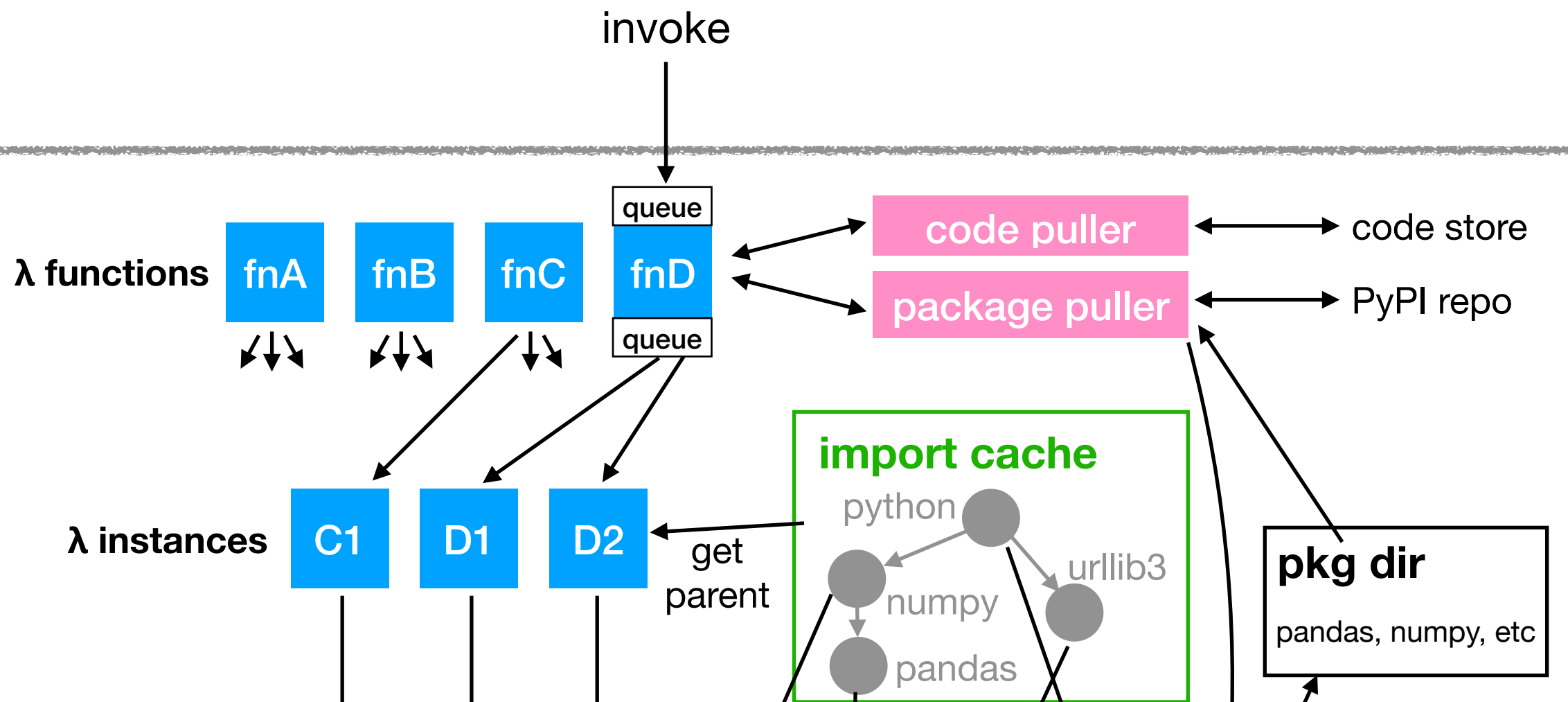
chroot

seccomp

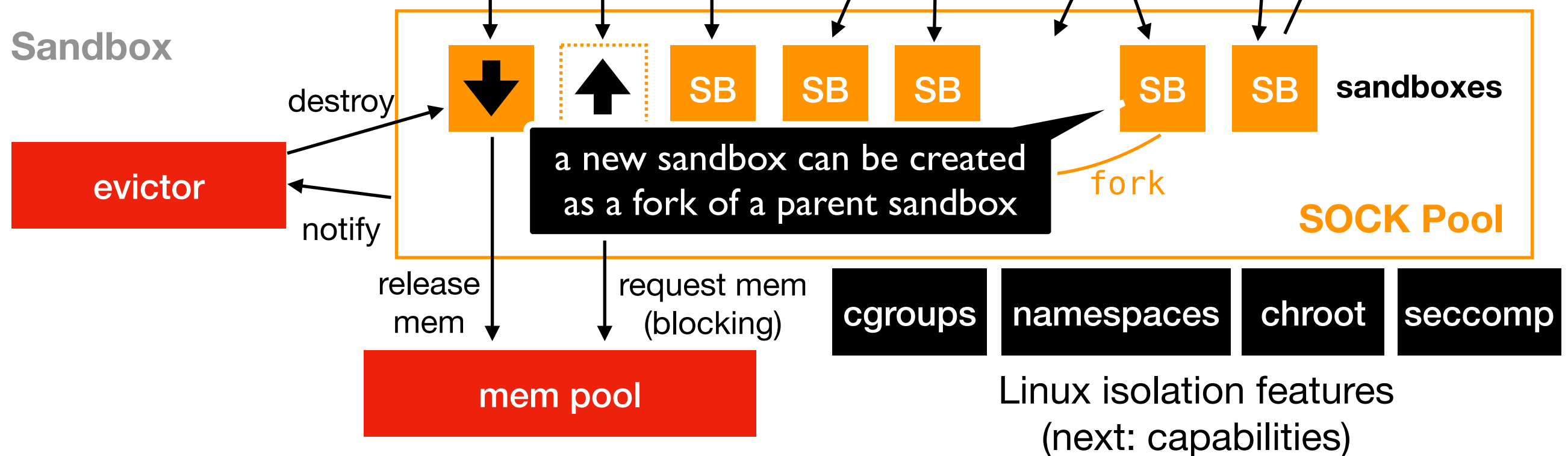
Linux isolation features
(next: capabilities)

Event

Lambda



Sandbox



```

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

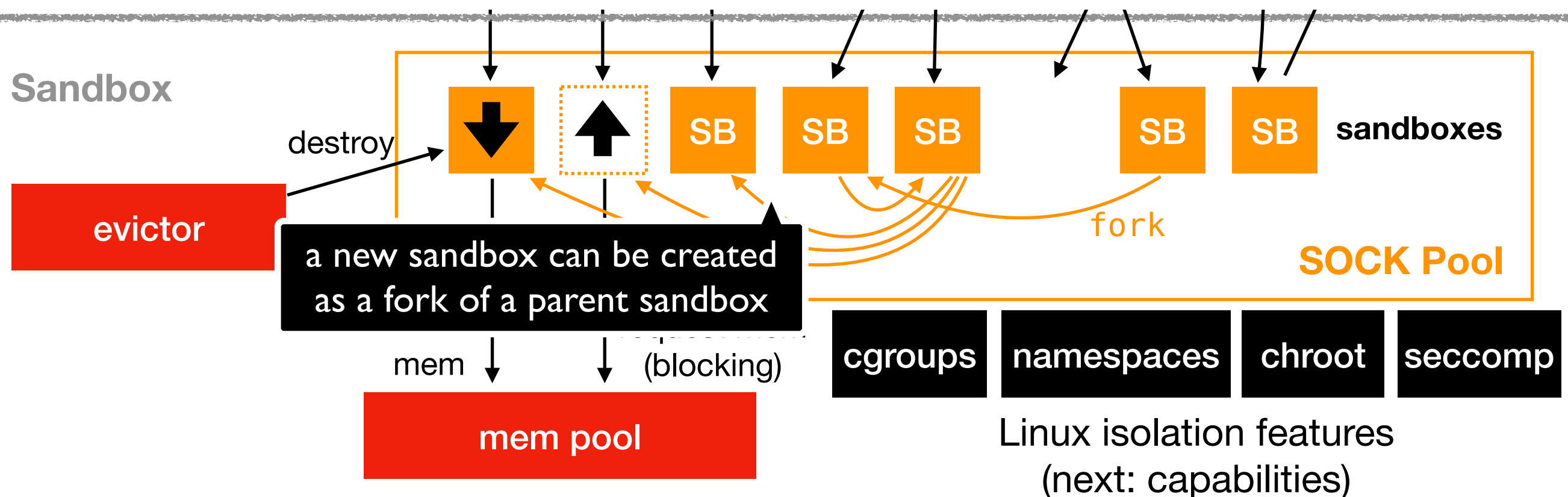
```

where to fork from
(may be null)

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

path to Lambda code

Sandbox API

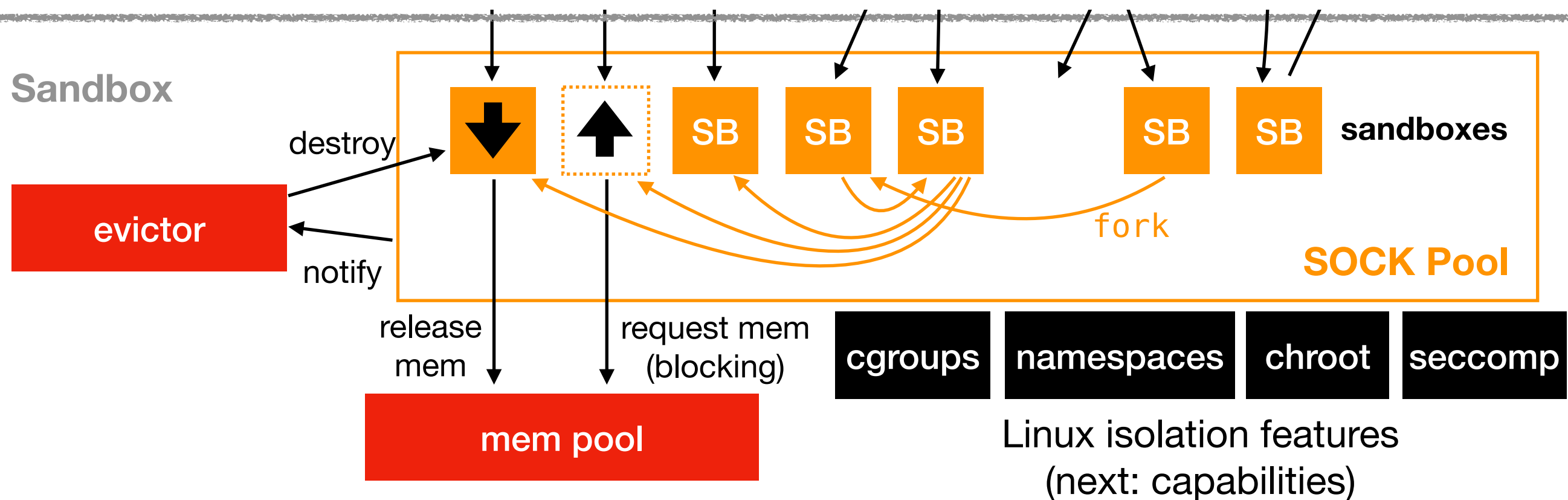


```

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)
}

```

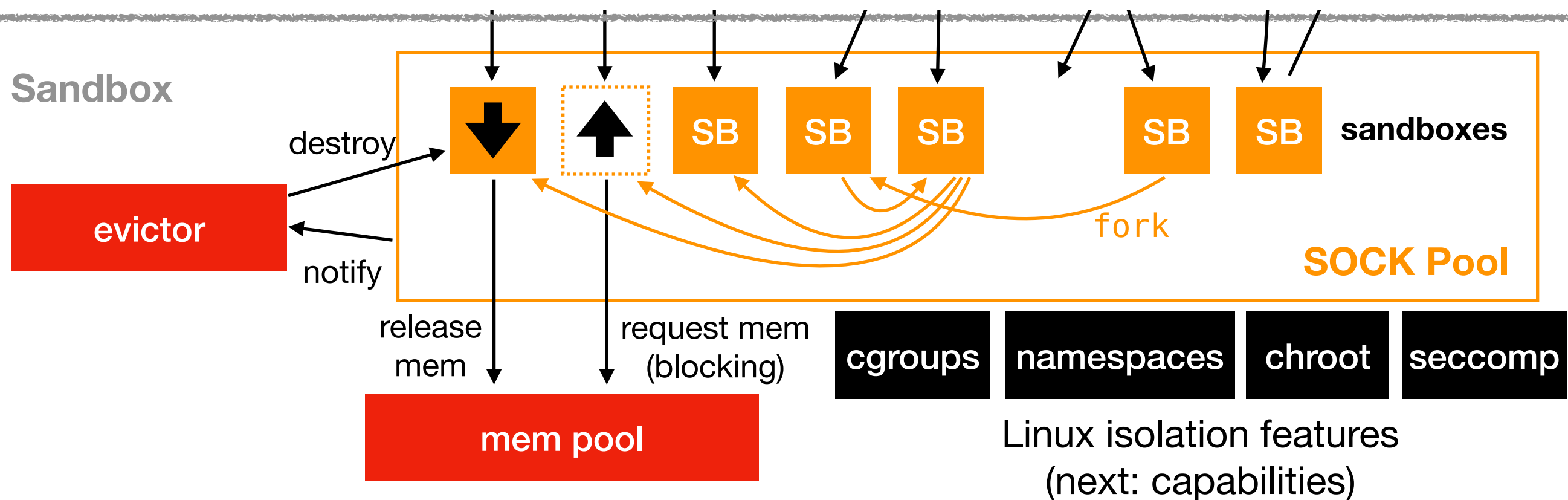


```

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) ← forward web requests
    Pause() error ← are Lambda processes
    Unpause() error ← schedulable by Linux?
    ...
    Destroy(reason string)
    DestroyIfPaused(reason string)
}

```

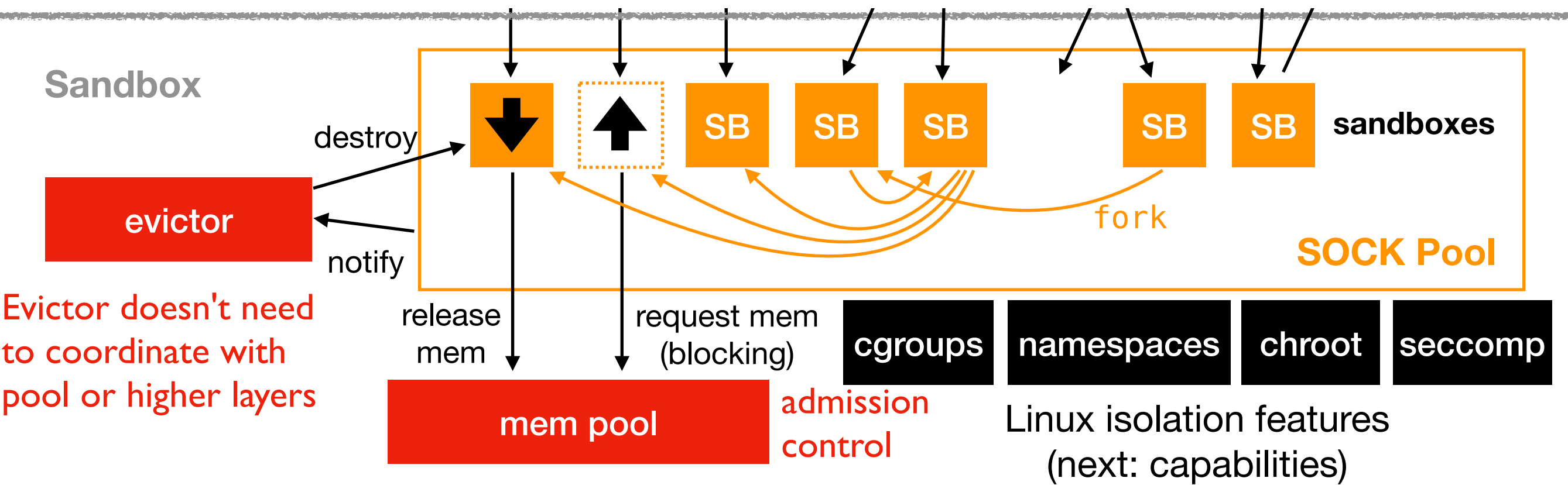



```
type safeSandbox struct {
    Sandbox

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

safeSandbox is a Sandbox and wraps every other Sandbox implementation

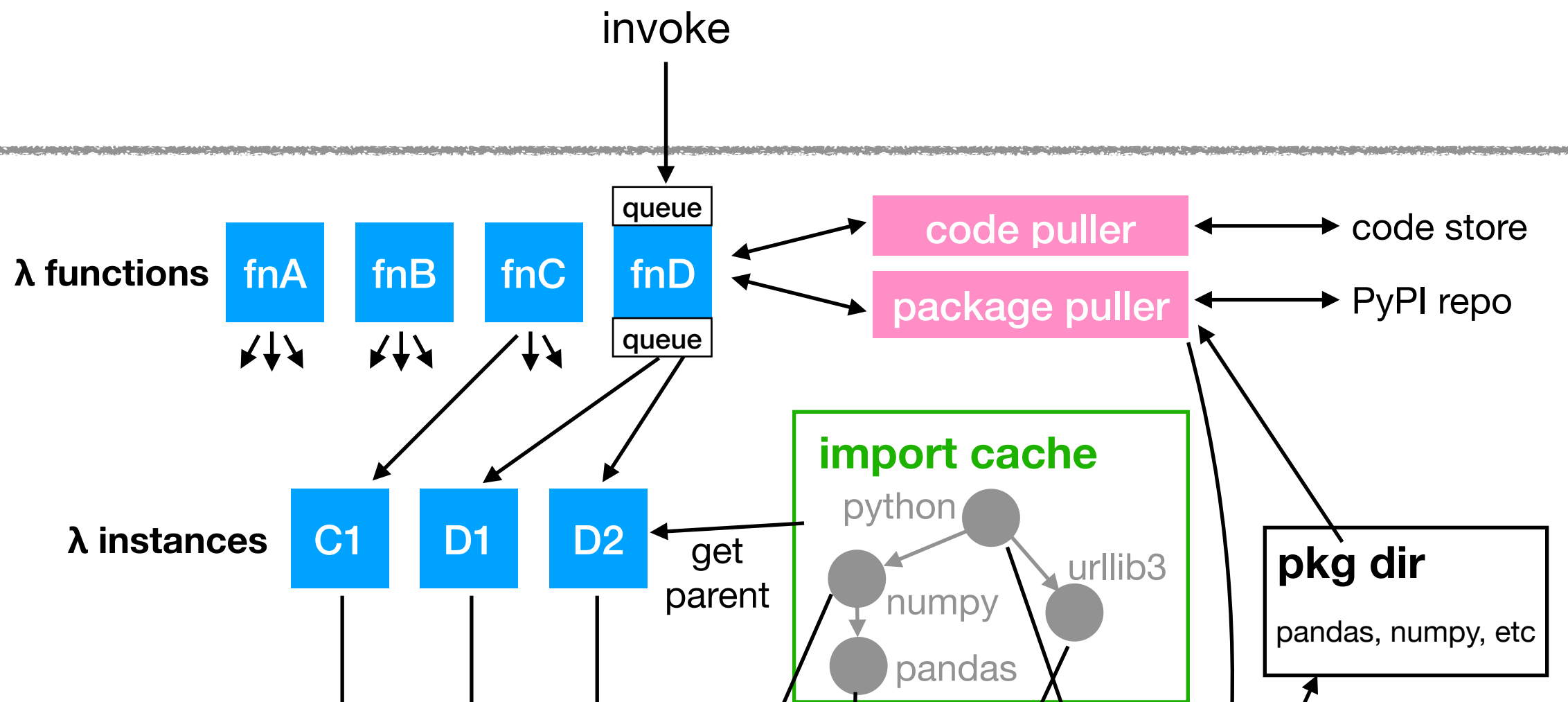
- Sandbox API is threadsafe
- If any call returns any error, Sandbox is automatically destroyed (caller can just stop using it)
- Any call on destroyed Sandbox returns error, causes no harm
- notifies listeners (particularly evictor) that can operate independently



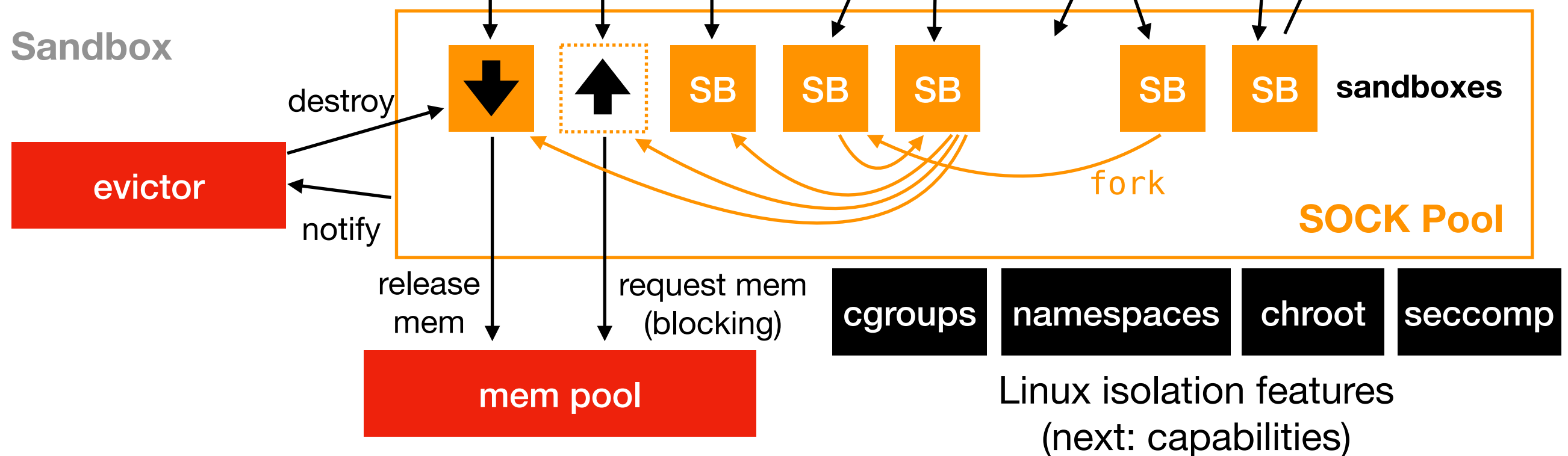
Evictor doesn't need to coordinate with pool or higher layers

Event

Lambda



Sandbox



Event

Lambda

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

numpy

pandas

urllib3

code puller

package puller

code store

PyPI repo

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.

evictor

dest

noti

release
mem

request mem
(blocking)

mem pool

cgroups

namespaces

chroot

seccomp

Linux isolation features
(next: capabilities)

SB

SB

sandboxes

ork

SOCK Pool

invoke

queue

ue

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

fnD

queue

queue

code puller

code store

package puller

PyPI repo

λ instances

C1

D1

D2

get parent

import cache

python

numpy

pandas

urllib3

pkg dir

pandas, numpy, etc

Sandbox

destroy

evictor

notify

release mem

request mem (blocking)

mem pool

SB

SB

SB

SB

SB

sandboxes

fork

SOCK Pool

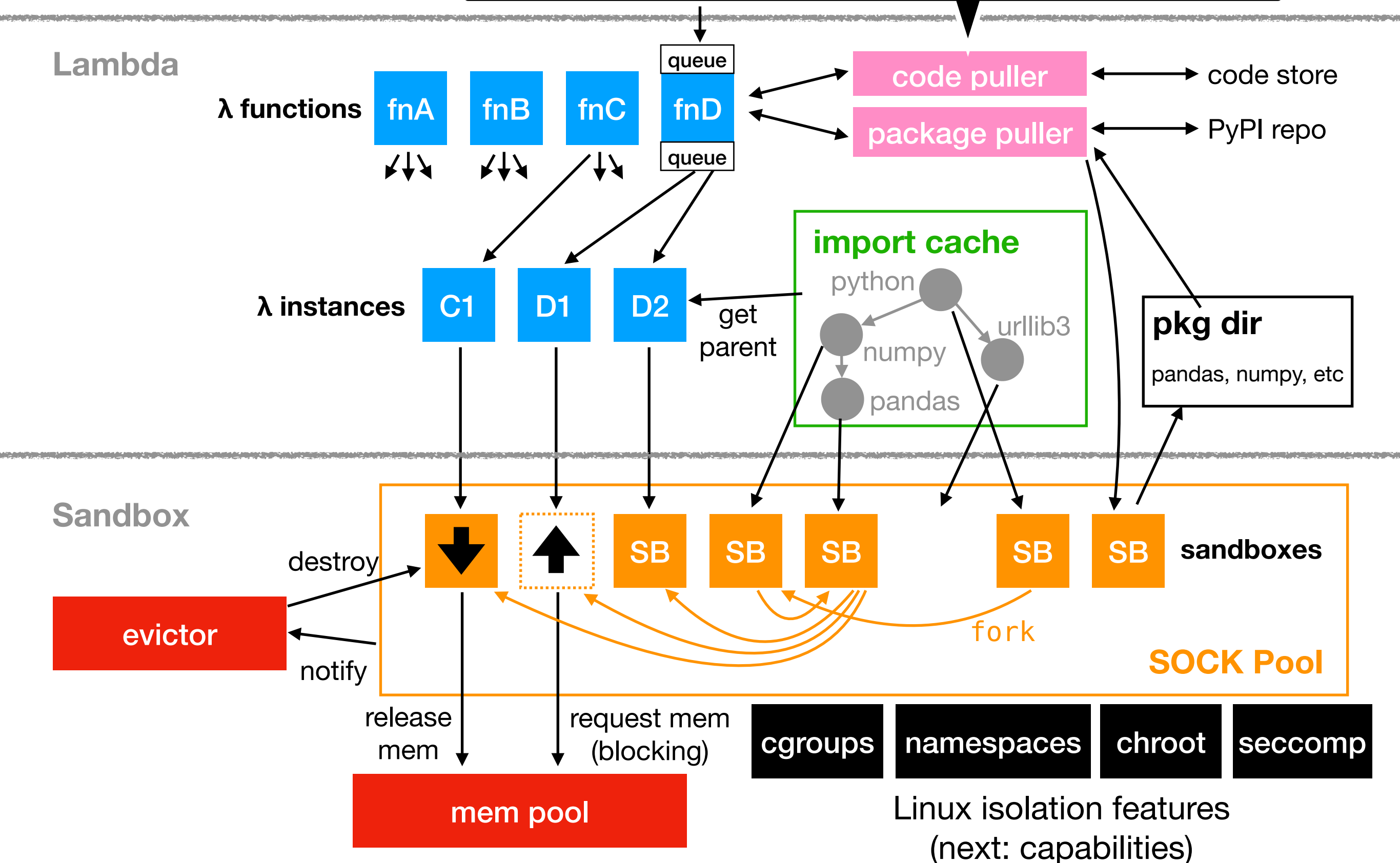
cgroups

namespaces

chroot

seccomp

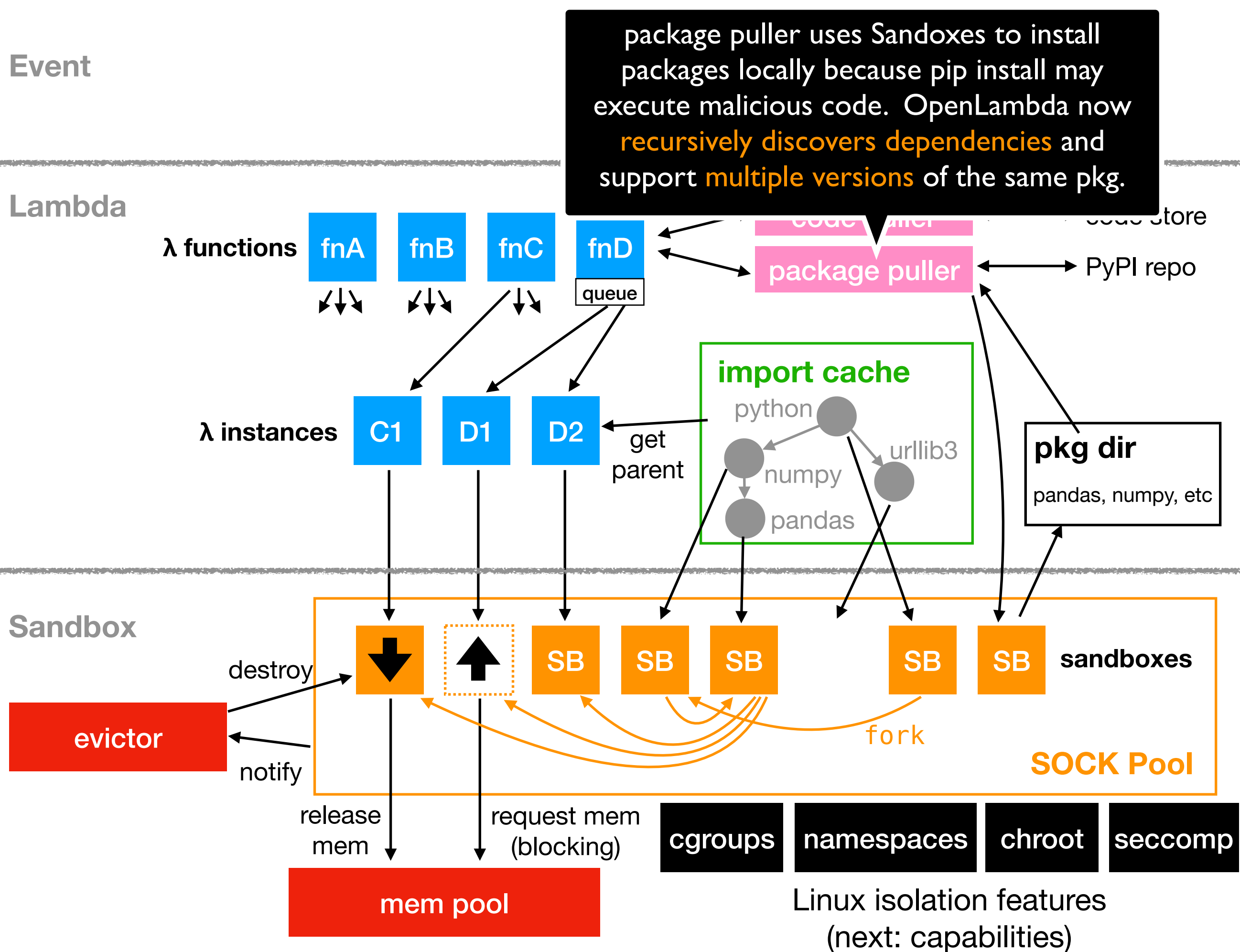
Linux isolation features
(next: capabilities)



Event

Lambda

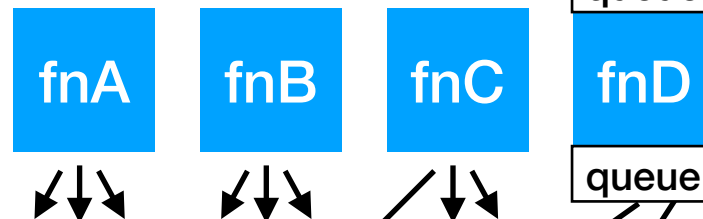
Sandbox



Event

Lambda

λ functions



- 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 instances

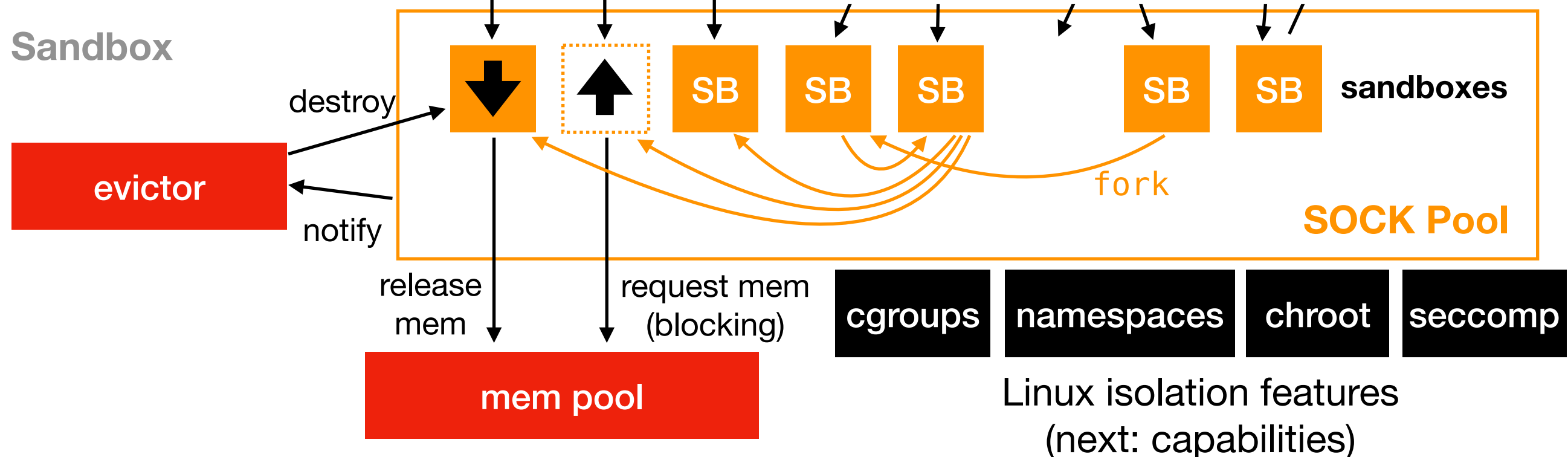


import cache

Basically a robust, **virtual Sandbox**

- sometimes backed by physical SB
- masks Sandbox failure by restarting them
- could broaden retry to handle failures within user code too

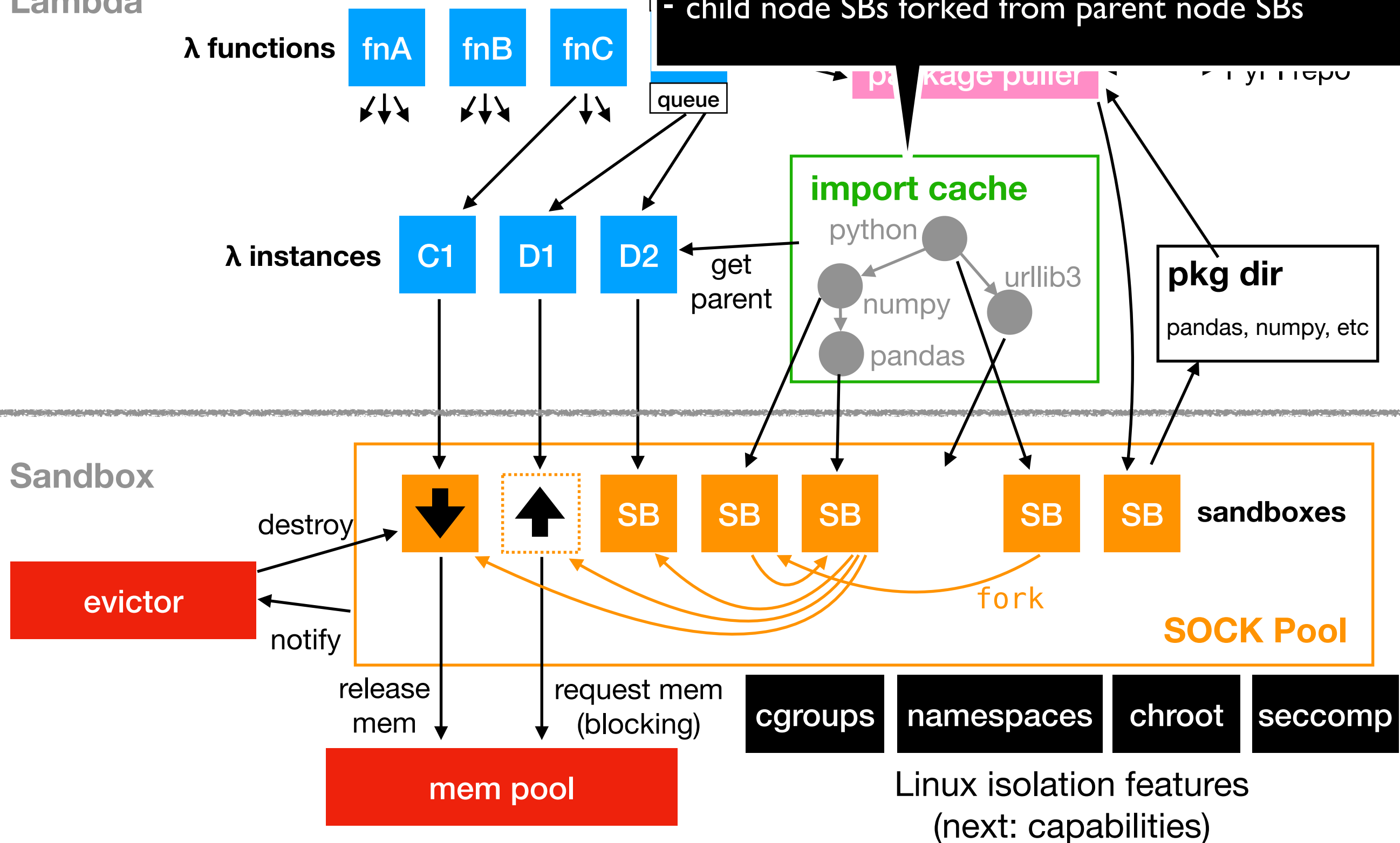
Sandbox



Event

Lambda

Sandbox



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

λ functions

fnC

```
queue
fnD
queue
```

code store

PyPI repo

C1

D1

D2

get
parent


python



urllib3

pkg dir

pandas, numpy, etc



SB



SB

SB

sandboxes

destroy

evictor

notify

release
mem

request mem
(blocking)



namespaces

chroot

seccomp

Linux isolation features (next: capabilities)

fork

SOCK Pool

