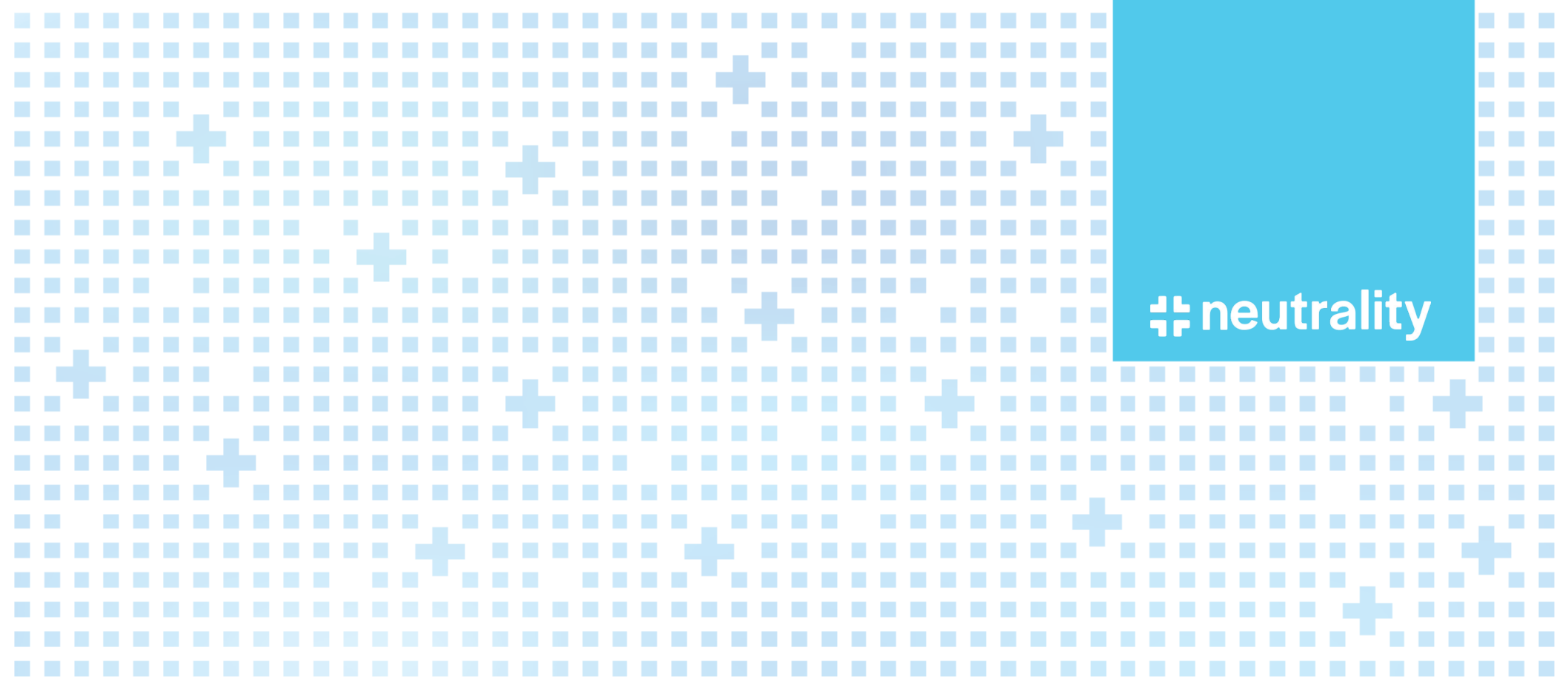


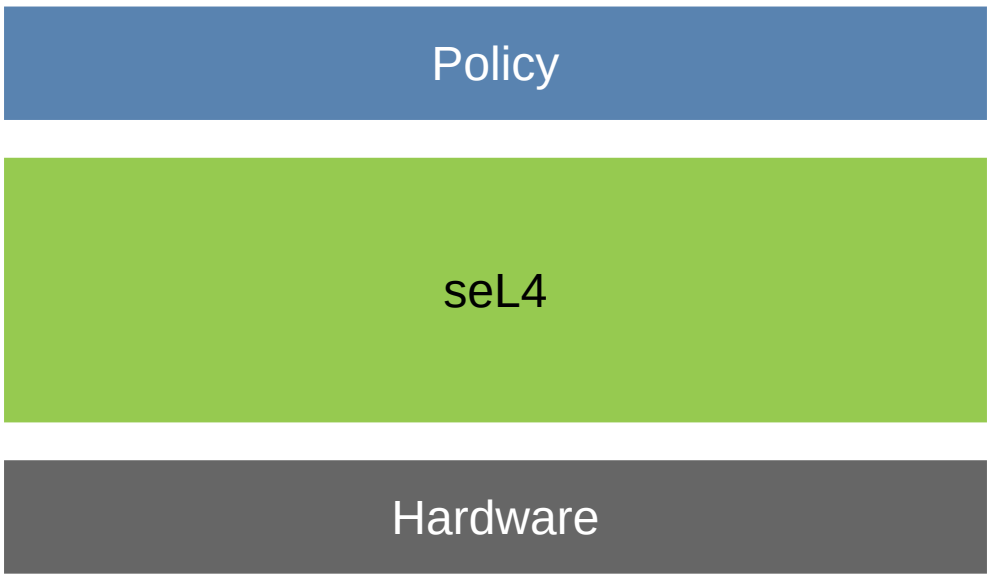
A Policy-free Boot Process for seL4

All our systems

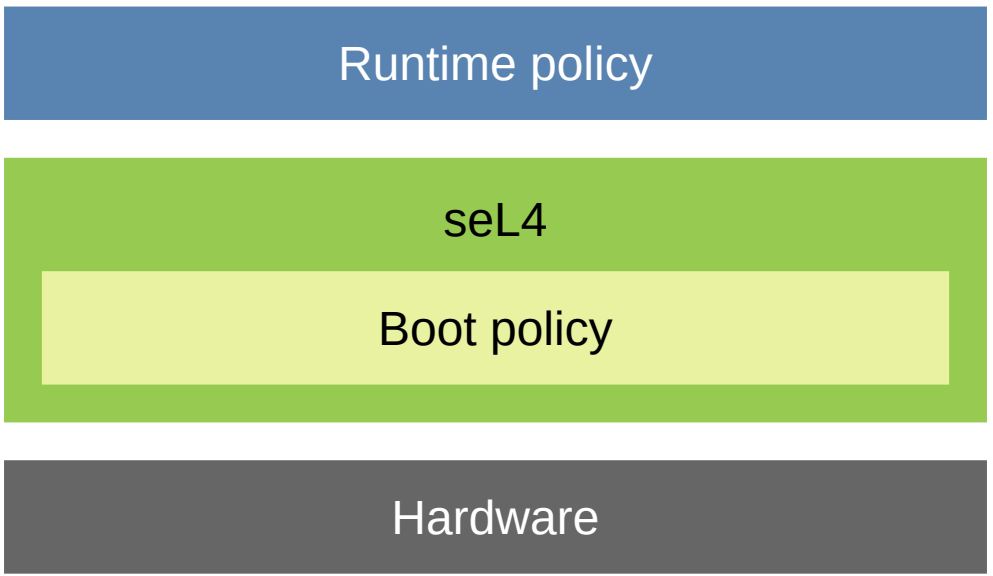
Hardware

neutrality

Bring your own policy ...

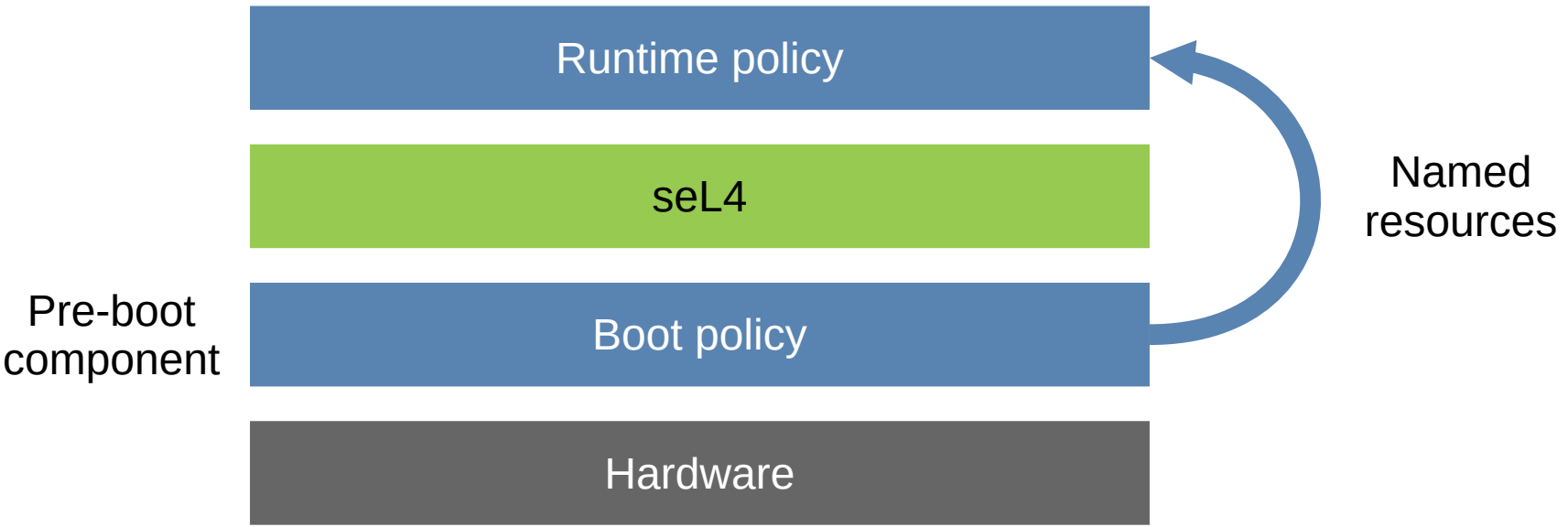


… except at boot



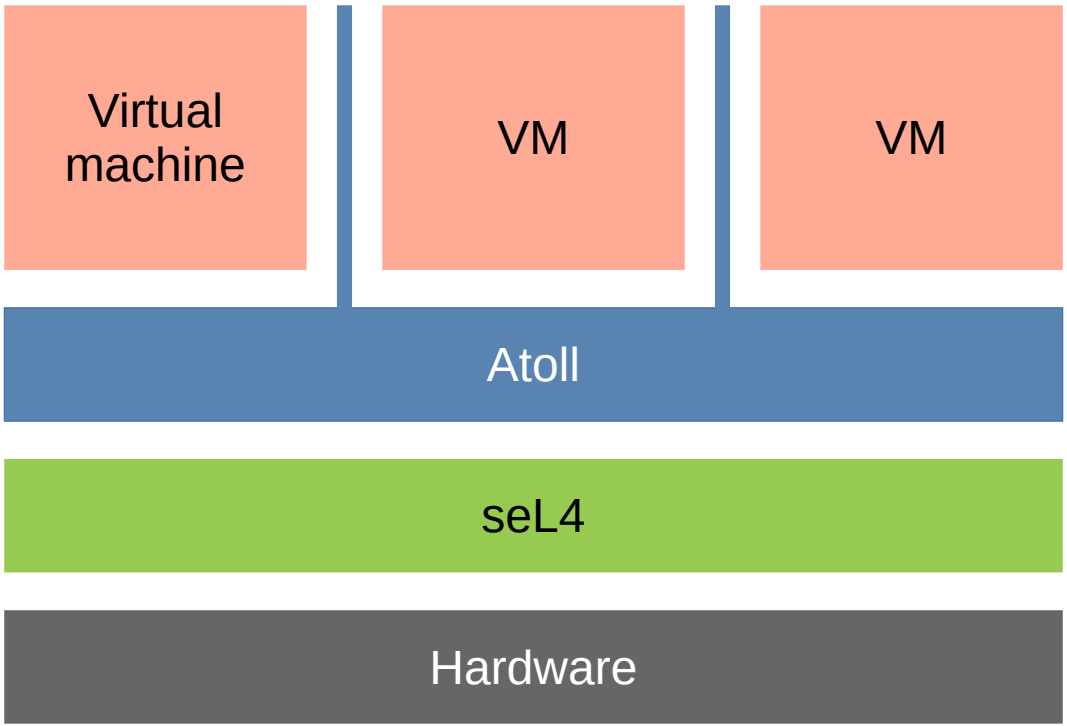
neutrality

Bring your own boot policy

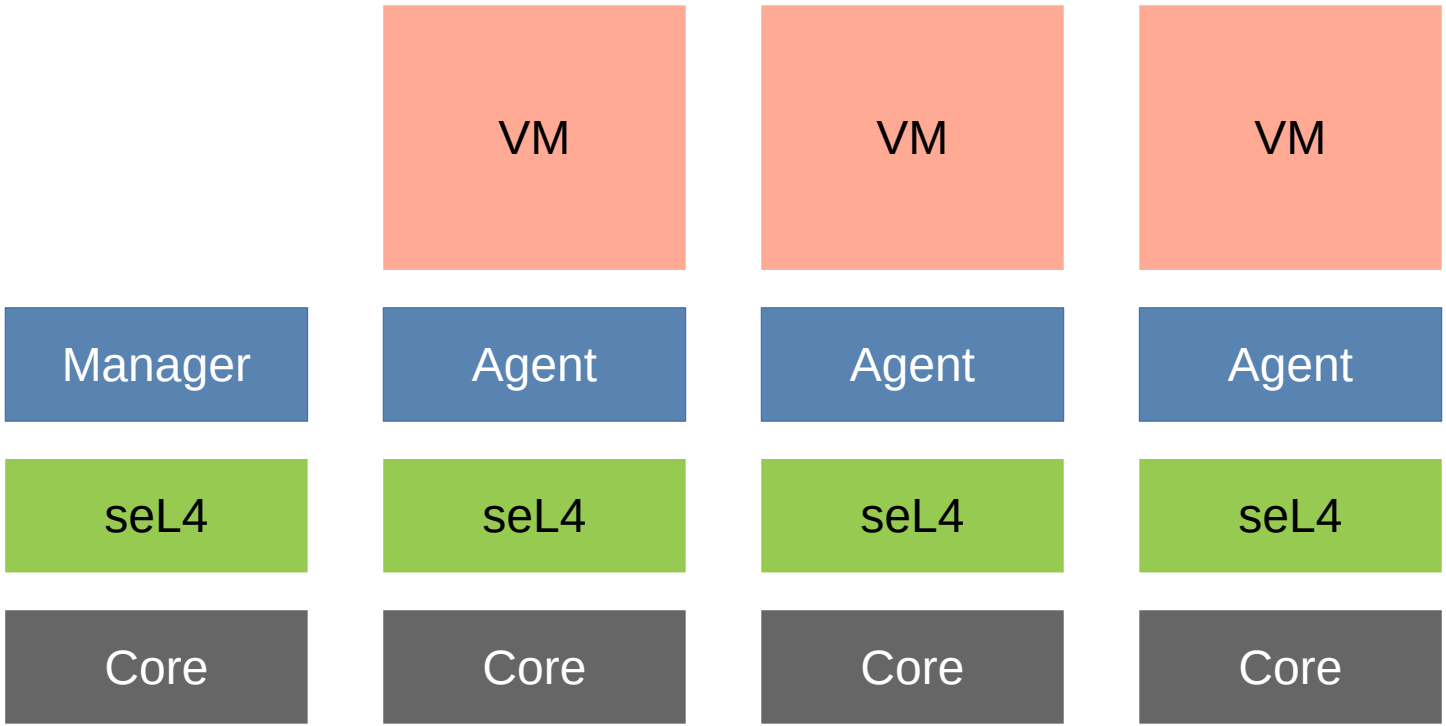


neutrality

Our System: Atoll



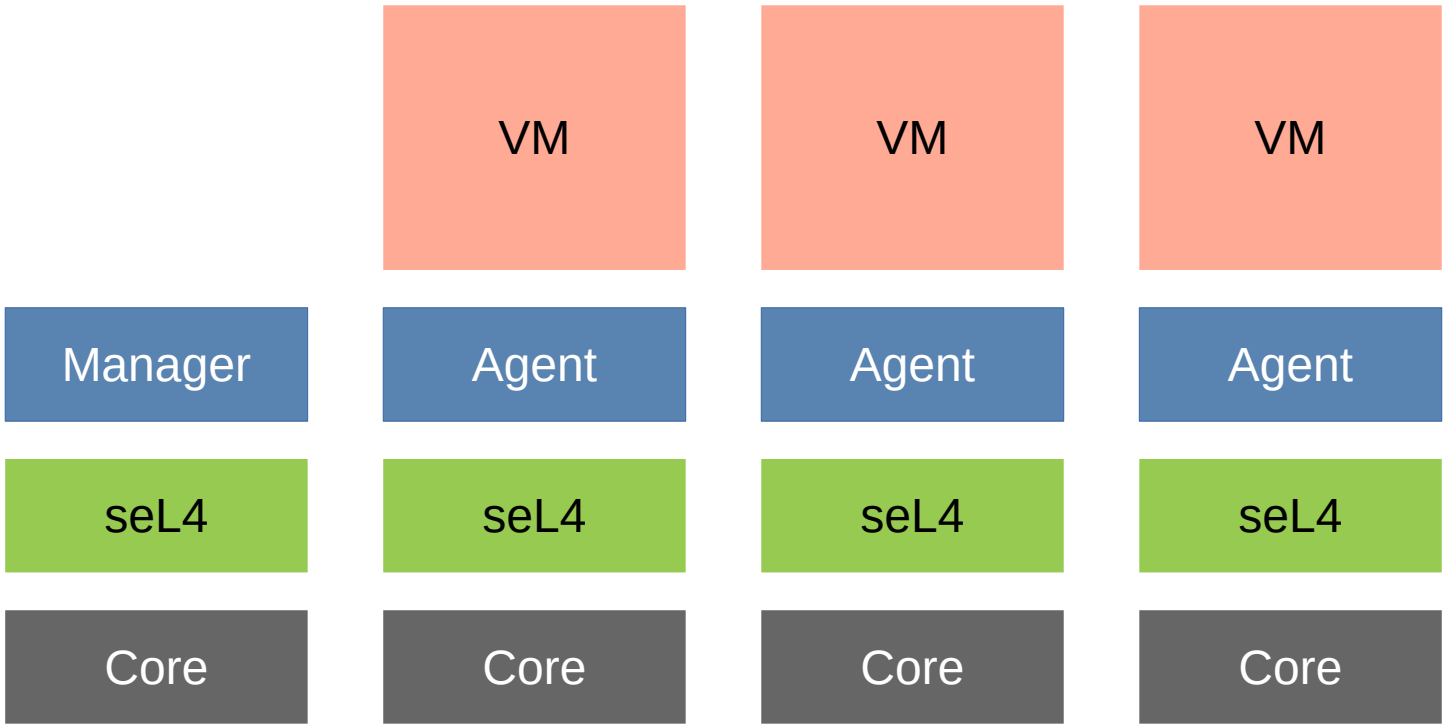
Atoll is a multikernel system



Breaking News

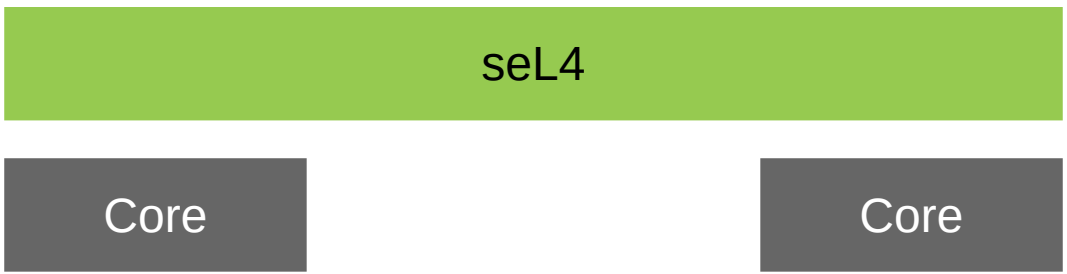
- Acquired European Innovation Council funding
- Grown to 9 full-time employees
- Multiple industrial pilots in ‘27

Atoll is a multikernel system

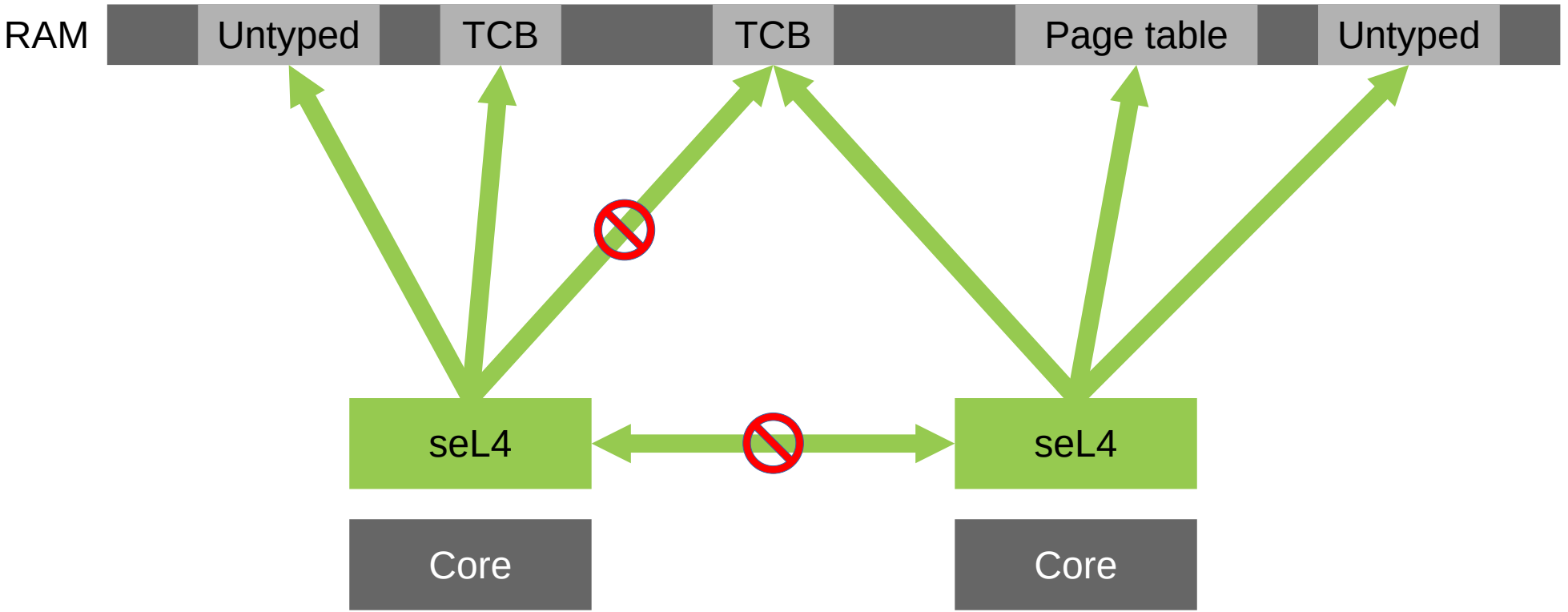


Multikernel 101

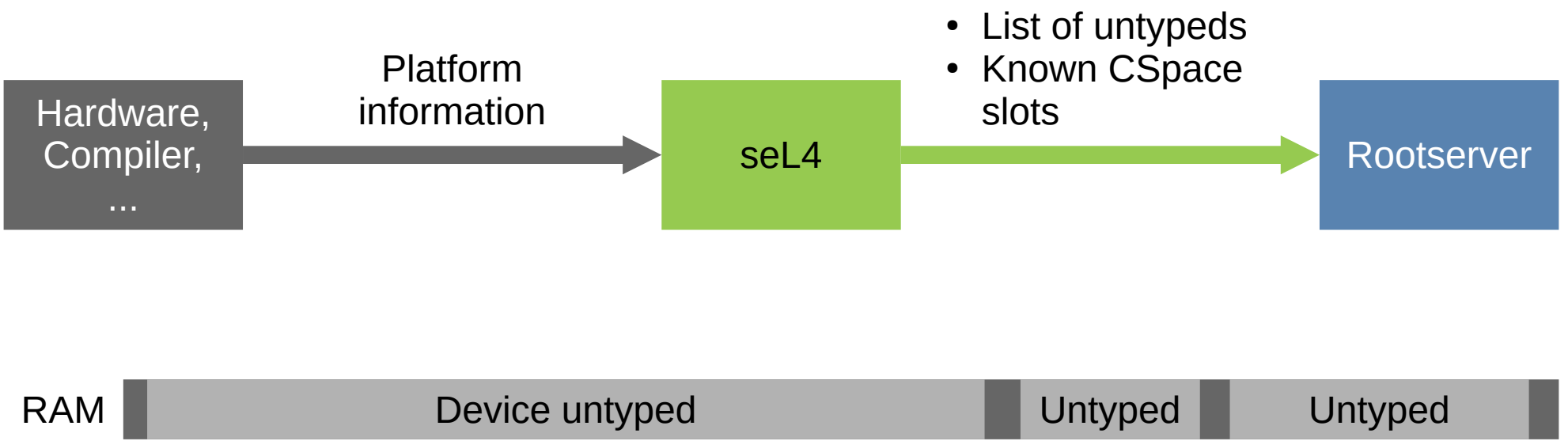
neutrality



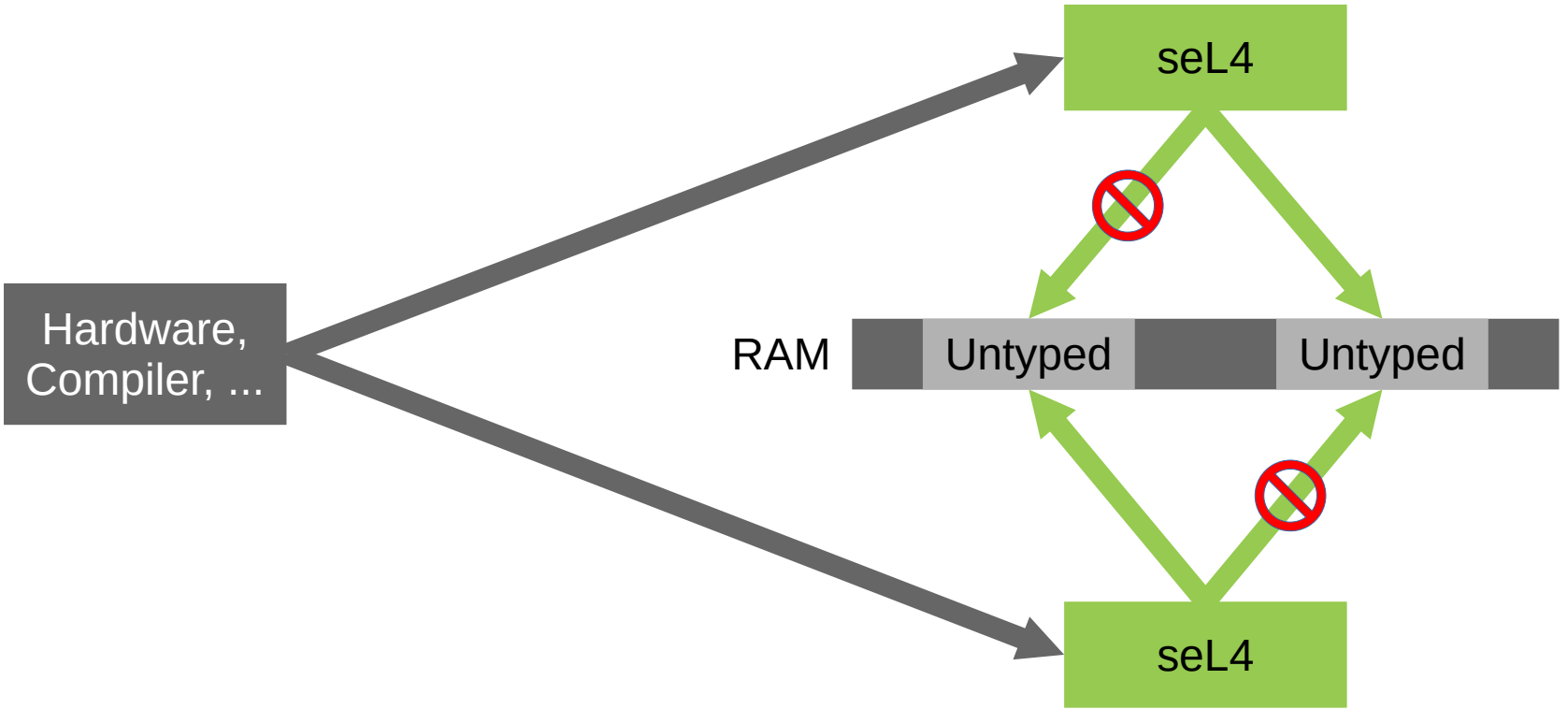
Multikernel 101



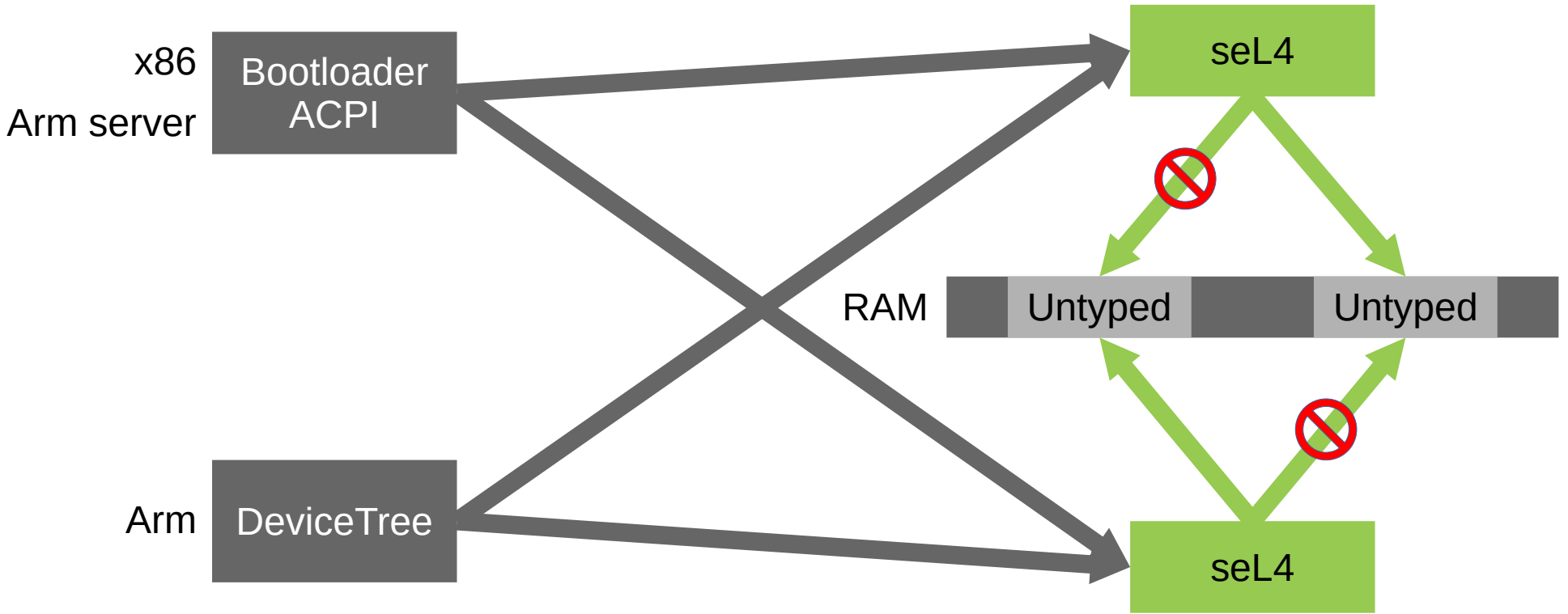
seL4 boot


Hardware, Compiler, ...
Platform information
seL4
• List of untypeds
• Known CSpace slots
Rootserver
RAM
Device untyped
Untyped
Untyped

Why fixed boot policy is a problem



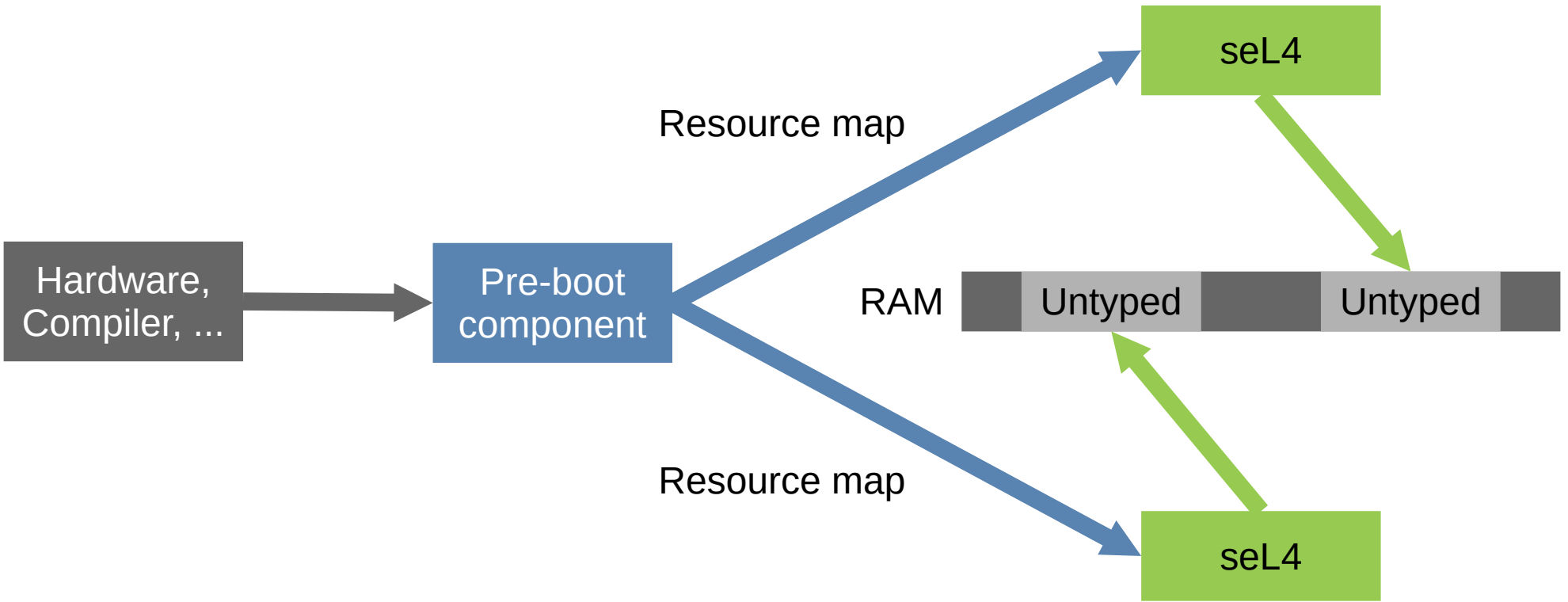
Why fixed boot policy is a problem



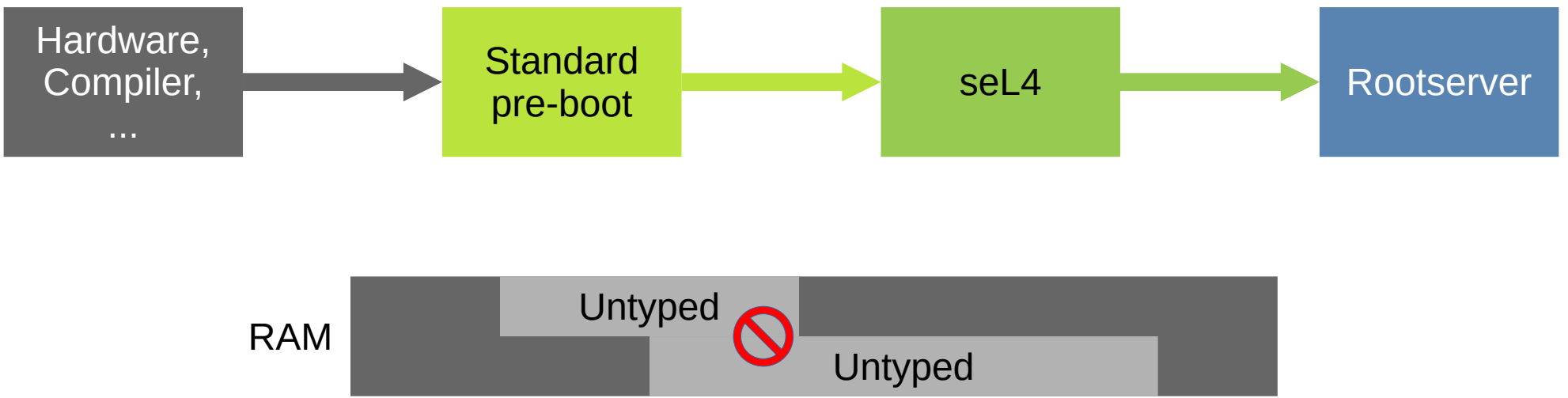
Boot code is part of the kernel

neutrality

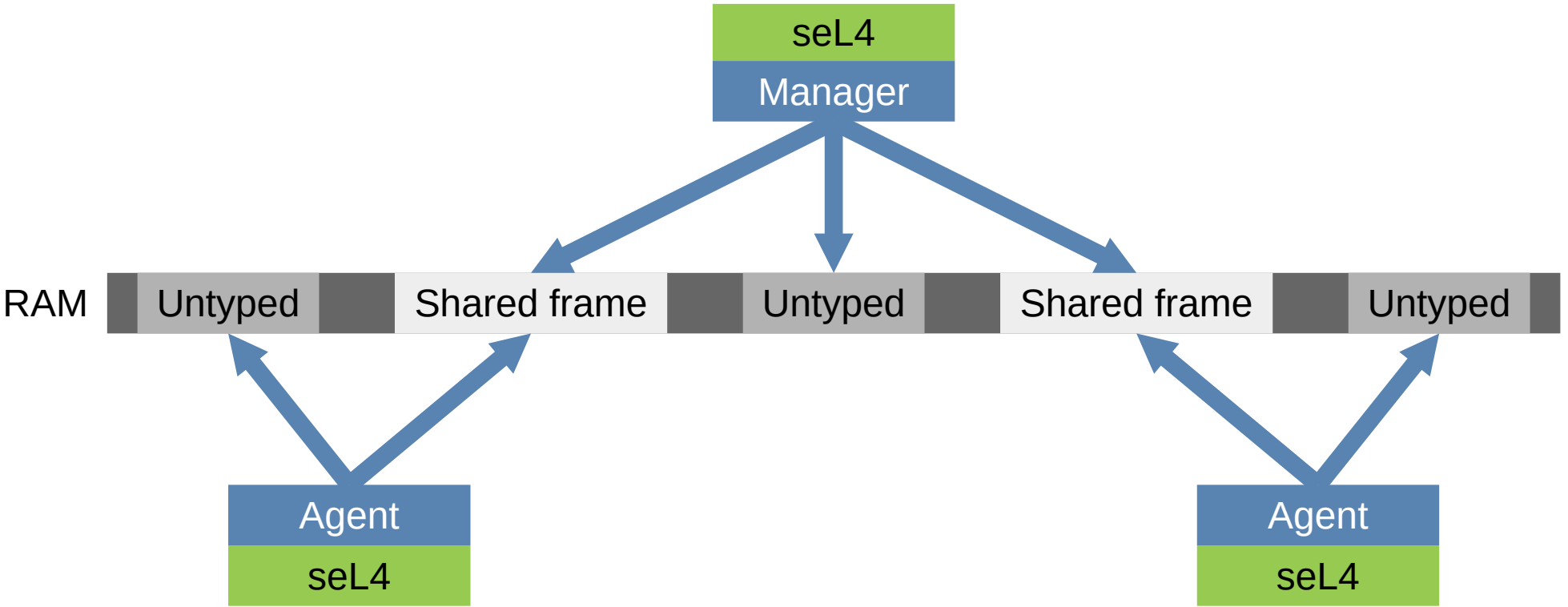
Externalize boot policy



Won’t this break everything?



Connecting pre-boot and runtime



neutrality

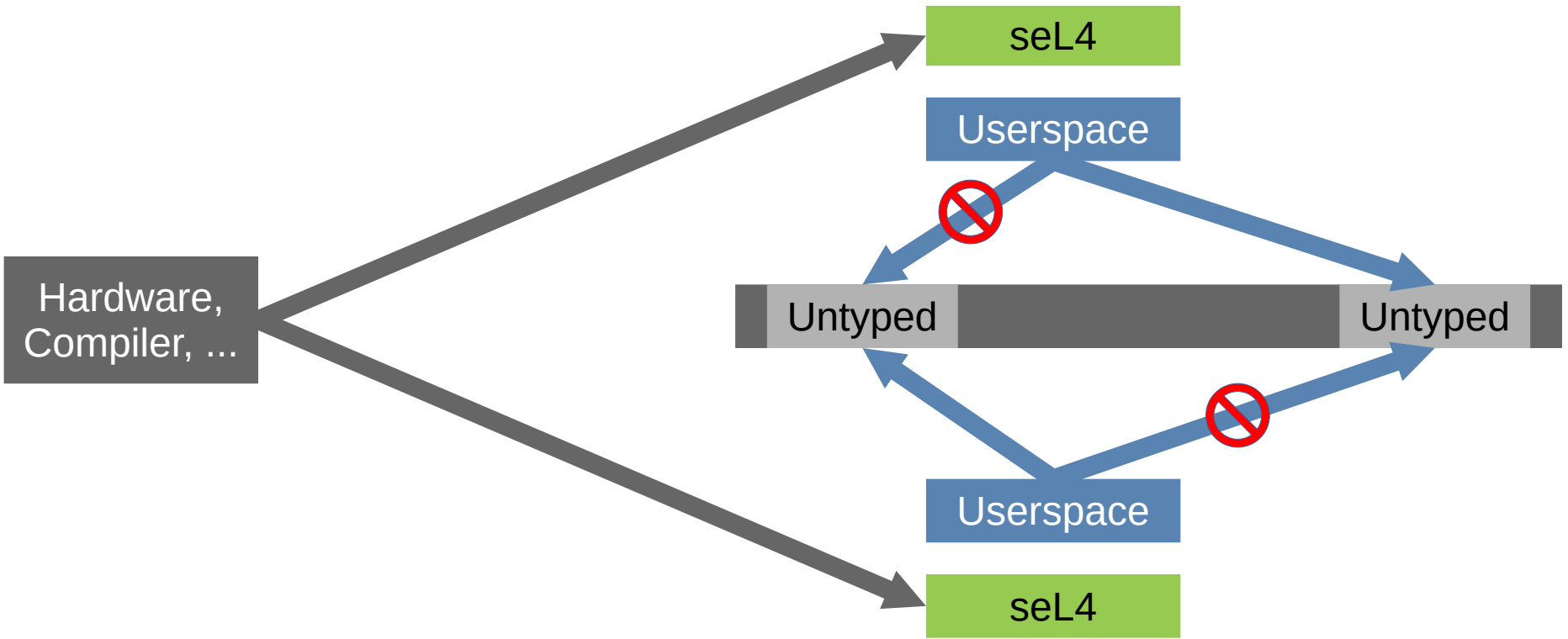
Named resources

```
/* Named resources are passed as an
additional bootinfo chunk */

typedef struct seL4_NamedResDesc {
  char name[32];
  object_t type;
  seL4_Uint8 sizeBits;
  seL4_SlotRegion slots;
} seL4_NamedResDesc;

typedef struct seL4_NamedResInfo {
  seL4_BootInfoHeader hdr;
  seL4_NamedResDesc namedRes[];
}
```

```
objects {
  queue_node1 = named_resource(
    AGENT_FRAME[0], frame
  )
  queue_node2 = named_resource(
    AGENT_FRAME[1], frame, optional
  )
  ...
}
caps {
  ...
  manager_pt {
      0x10: queue_node1 (RW)
      0x20: queue_node2 (RW)
  }
}
```

neutrality

Booting the microkernel way



neutrality

Booting the microkernel way