code

map

P1   P2   P2

code
heap
map
stack

mem { code
heap
stack }

3 .bpf.c

ELF types
RPC

fresh init regs

Map

map
normal data
code1   code2

storage server aka Map

>> Serverless functions?!

for al
map   ep2 code2
ep1 code1

storage server "map"
get/set

R1 code ep stack   R2

persist

1. jump into Rust code multiple ep

2. return to a fresh state

storage
R1   R2

export for fev

heap/
"globals" ← actually
a map

"bpf programs
=
distributed
serverless
system"

↓
long-lived
programs

snapshot/restore

code

→ not just composing functions, but also kernel!

→ bpf gets around "slow start" problems because
it's 100% stateless
& preloaded

which more general
progs may
not be