Commoditize Your CI Compute: Spot Instances Without the Spotty Reliability

Felicitas Pojtinger (maintainer of Drafter)
@pojntfx

Loophole Labs

Are you running your CI/CD on spot instances?

Pay only for
what you use

70-90%
savings

Better for
cloud
providers, too

30-120s termination notice

Loophole Labs

❌ **Expensive** on-demand instances

❌ Spots only for **stateless** jobs

❌ Complex custom **checkpointing**

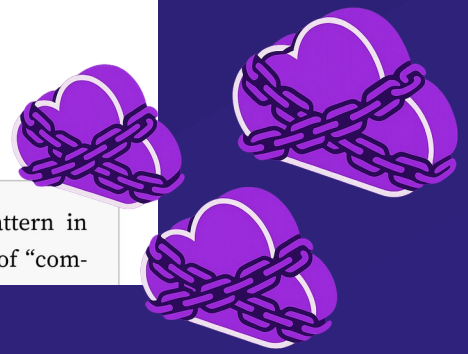
❌ Accept failures and **retry**

Fewer tests that don’t run for each commit

Short or no **benchmarks**

€$¥

No or little **fuzzing**



ABOUT SITE | ABOUT ME | NEW ESSAYS | NEW LINKS | PATREON

Laws of Tech: Commoditize Your Complement

AI economics, *economics*, *insight porn*, *Google*

A classic pattern in technology economics, identified by Joel Spolsky, is layers of the stack attempting to become monopolies while turning other layers into perfectly-competitive markets which are commoditized, in order to harvest most of the consumer surplus; discussion and examples.

2018-03-17–2022-01-11 (3y) · *finished* · certainty: *highly likely* · importance: *5*

backlinks · similar · bibliography

1 **"Smart Companies Try To Commoditize Their Products' Complements"**

Joel Spolsky in 2002 (23ya) identified a major pattern in technology business & economics: the pattern of "com-

https://gwern.net/complement

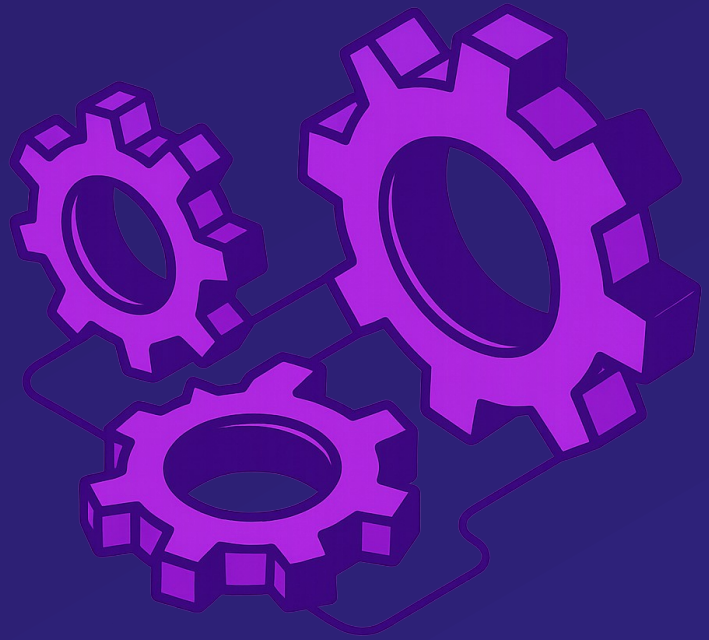
architect

“A unified, failure-proof compute fabric”

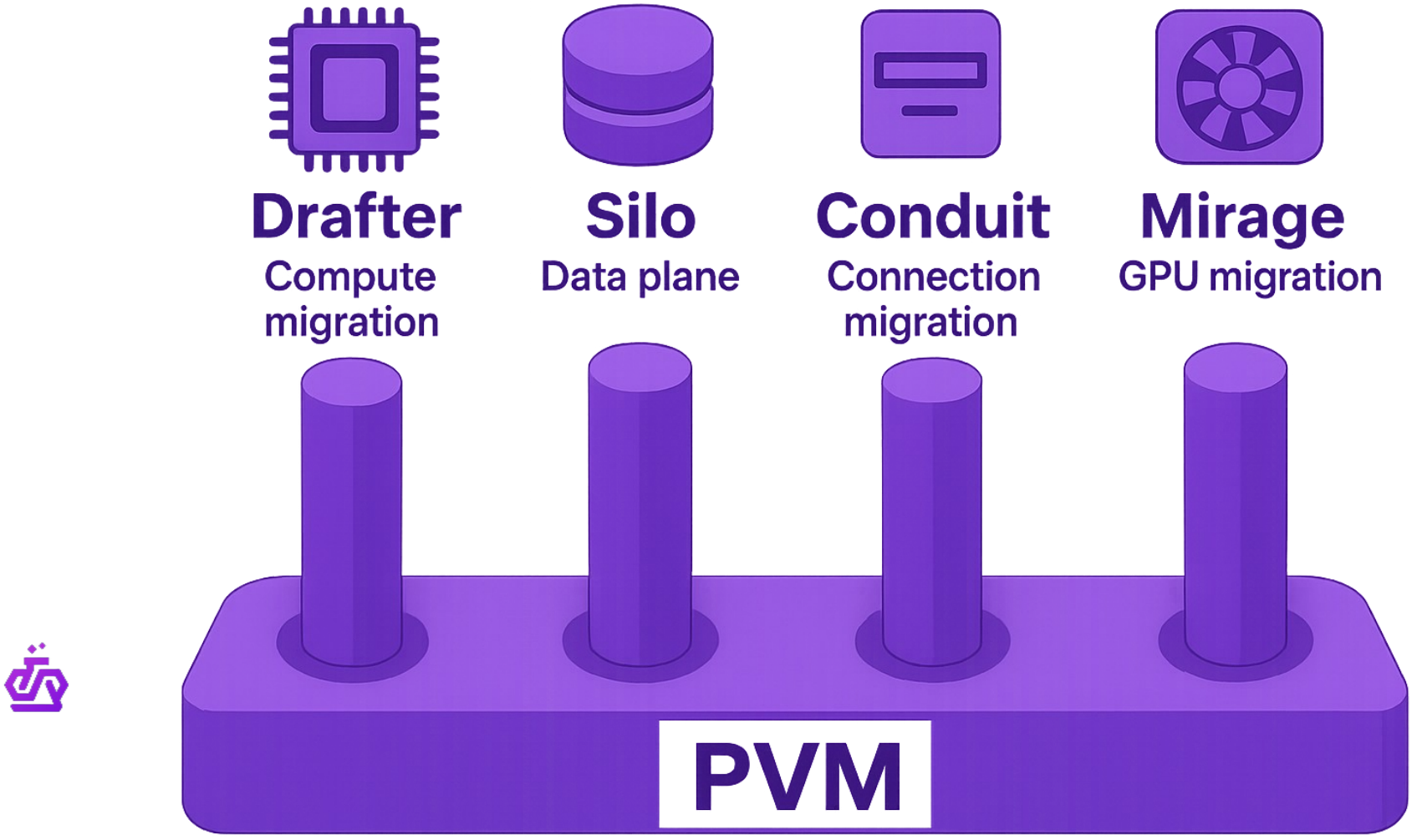
Loophole Labs

Cloud-agnostic (AWS, GCP, Azure, on-prem etc.) VM images and snapshots

Very fast **live migration**

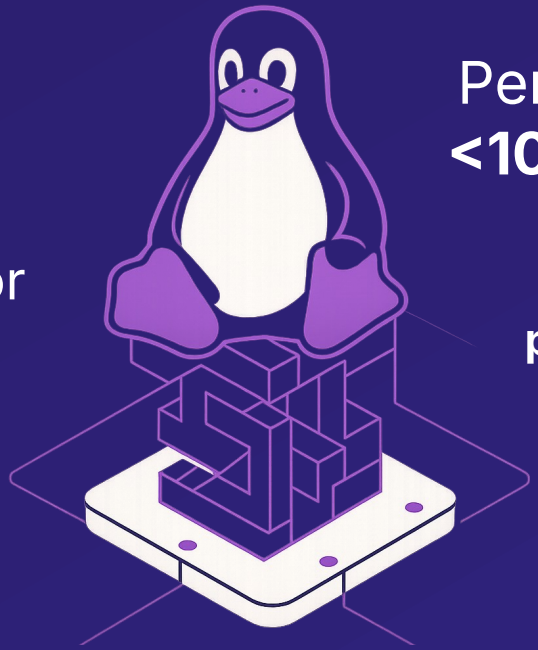


Complete **compliance**


Drafter
Compute migration
Silo
Data plane
Conduit
Connection migration
Mirage
GPU migration
PVM

PVM: Pagetable Virtual Machine

Run VMs on VMs
without nested virtualization or emulation



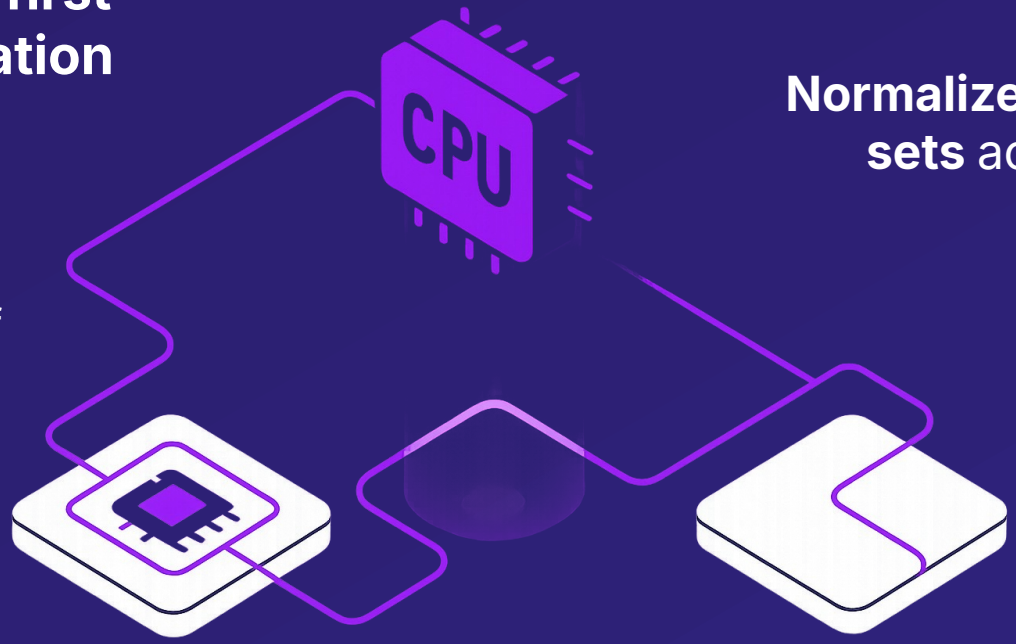
Performance **overhead**
<10% compared to host

Custom Loophole Labs patches for live migration

Drafter: Compute Unit

Hypervisor with **first-class live migration** support

Based on our fork of **Firecracker**



Normalizes CPU feature sets across hosts

Silo: Data Migration System

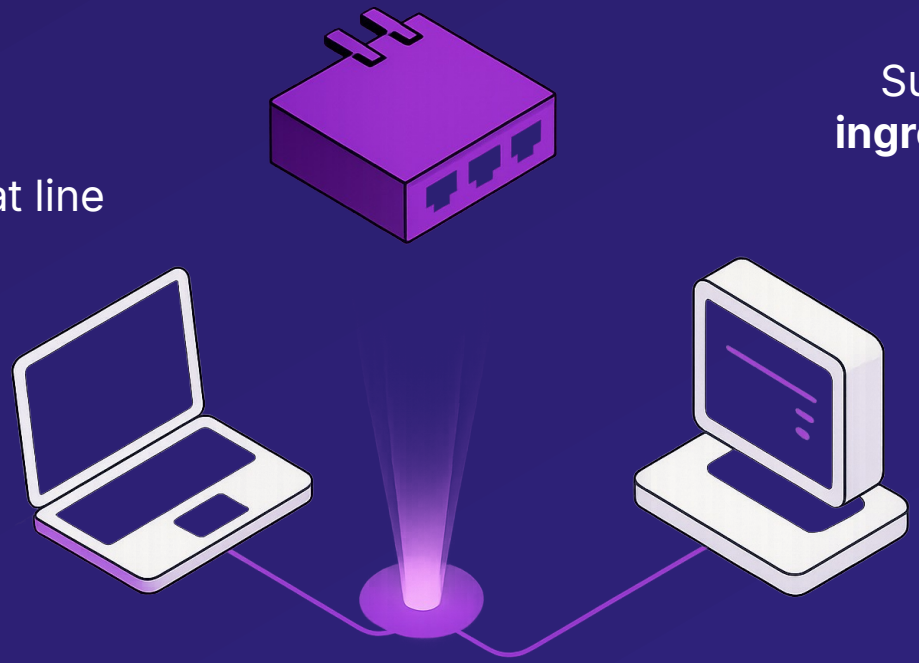
Tracks disk & memory changes and learns **access patterns**

Prioritizes important blocks over P2P connection and falls back to S3

Continuously backs up to S3

Conduit: Network Migration

Written in **eBPF**:
Runs on network card at line speed



Supports migrating
ingress & egress traffic

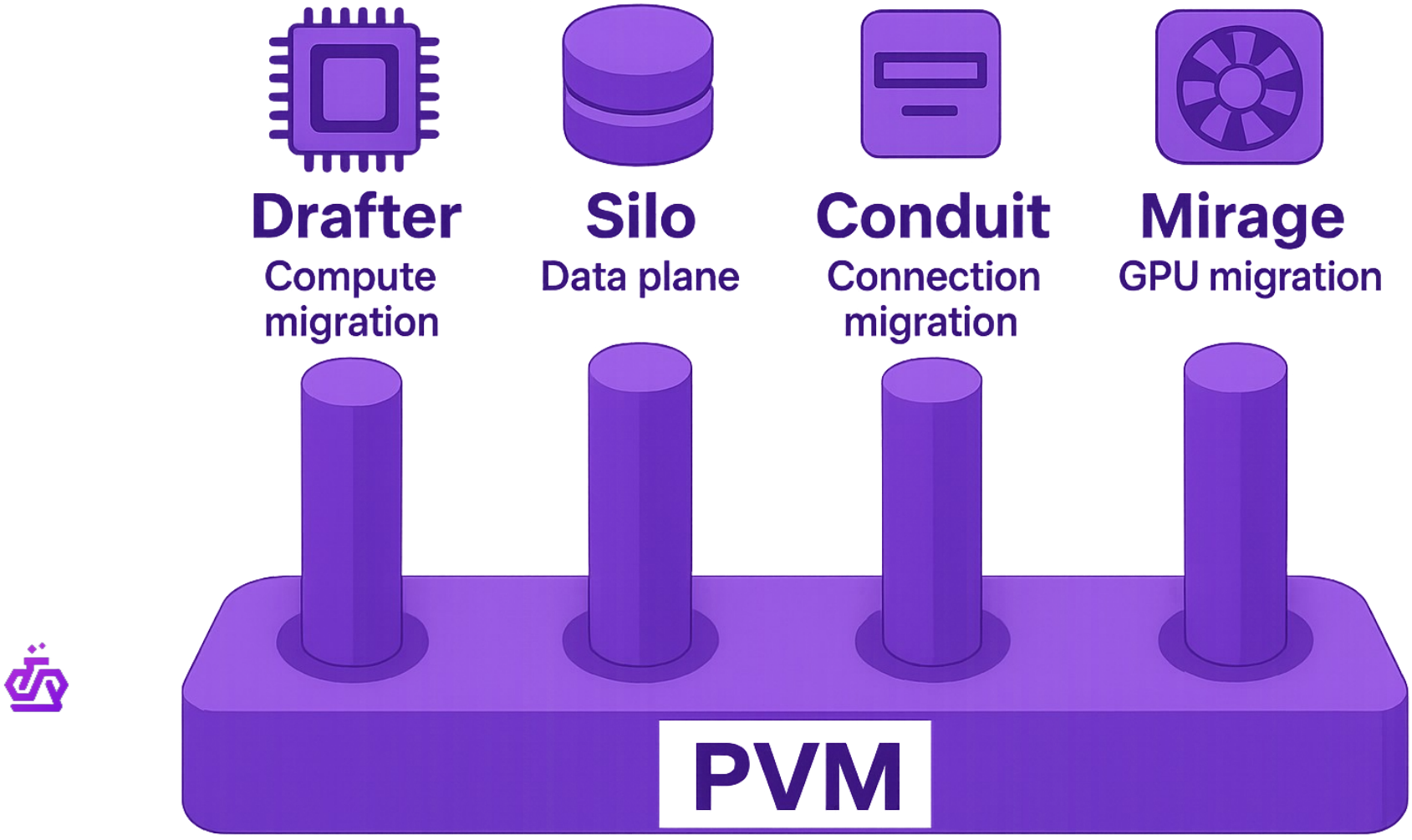
Works on L3 (both UDP & TCP are supported)

Mirage: GPU Workload Migration

Zero application changes needed

Supports both **inference and training**

Supports migrating a workload between **heterogeneous hosts and GPUs**


Drafter
Compute migration
Silo
Data plane
Conduit
Connection migration
Mirage
GPU migration
PVM

Orchestrates Drafter, Silo, Conduit and Mirage

Provisions physical instances from cloud providers

Exposes **GitHub actions**, **Kubernetes** and **REST API** integrations

Rapidly evacuate from instances after termination notice

Migrate closer to your users as access patterns change

Cross-cloud arbitrage: Automatically migrate to cheapest provider

Demo:

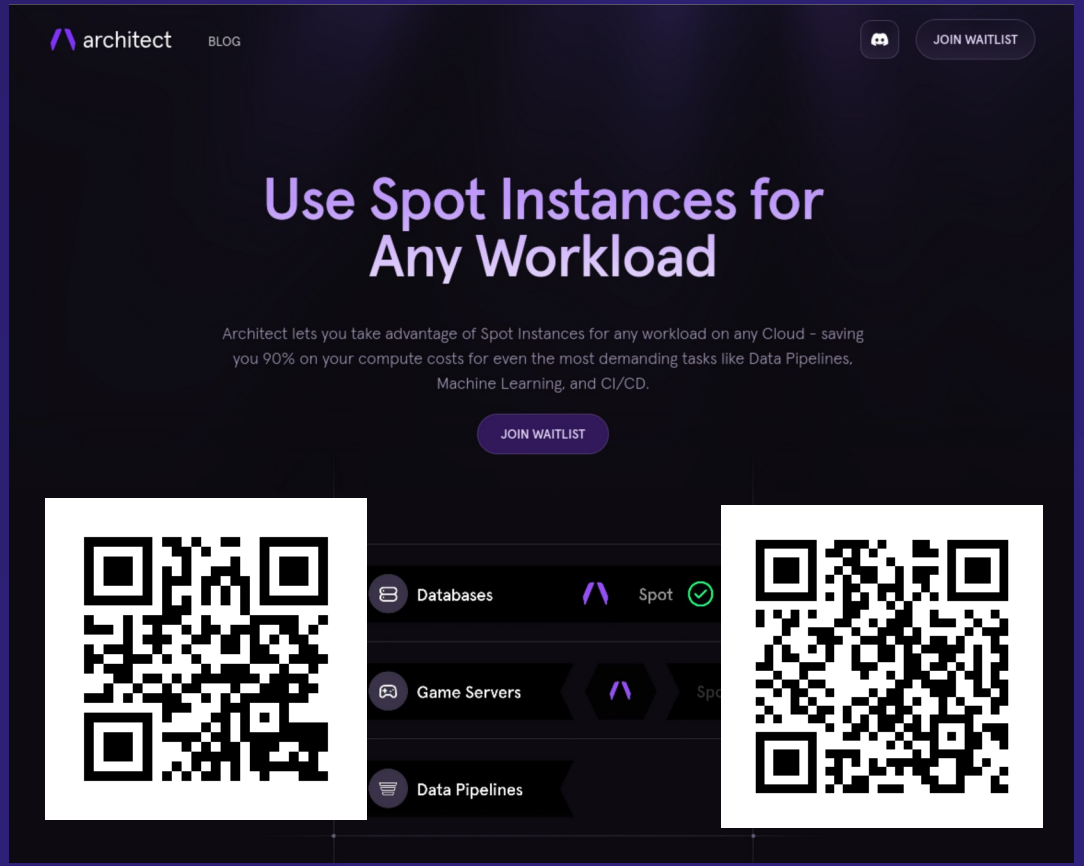
Live Migrating a Fuzzing Workload Across Cloud Providers Using Architect

Loophole Labs

Felicitas Pojtinger

Fediverse: @pojntfx@mastodon.social
Bluesky: @felicitas.pojtinger.com
Github: @pojntfx
LinkedIn: in/pojntfx
Web: felicitas.pojtinger.com

Loophole Labs



Waitlist:
https://architect.run/

Links & Resources:
https://loophole.sh/bugbash2025