
KubeCon
CloudNativeCon
North America 2023

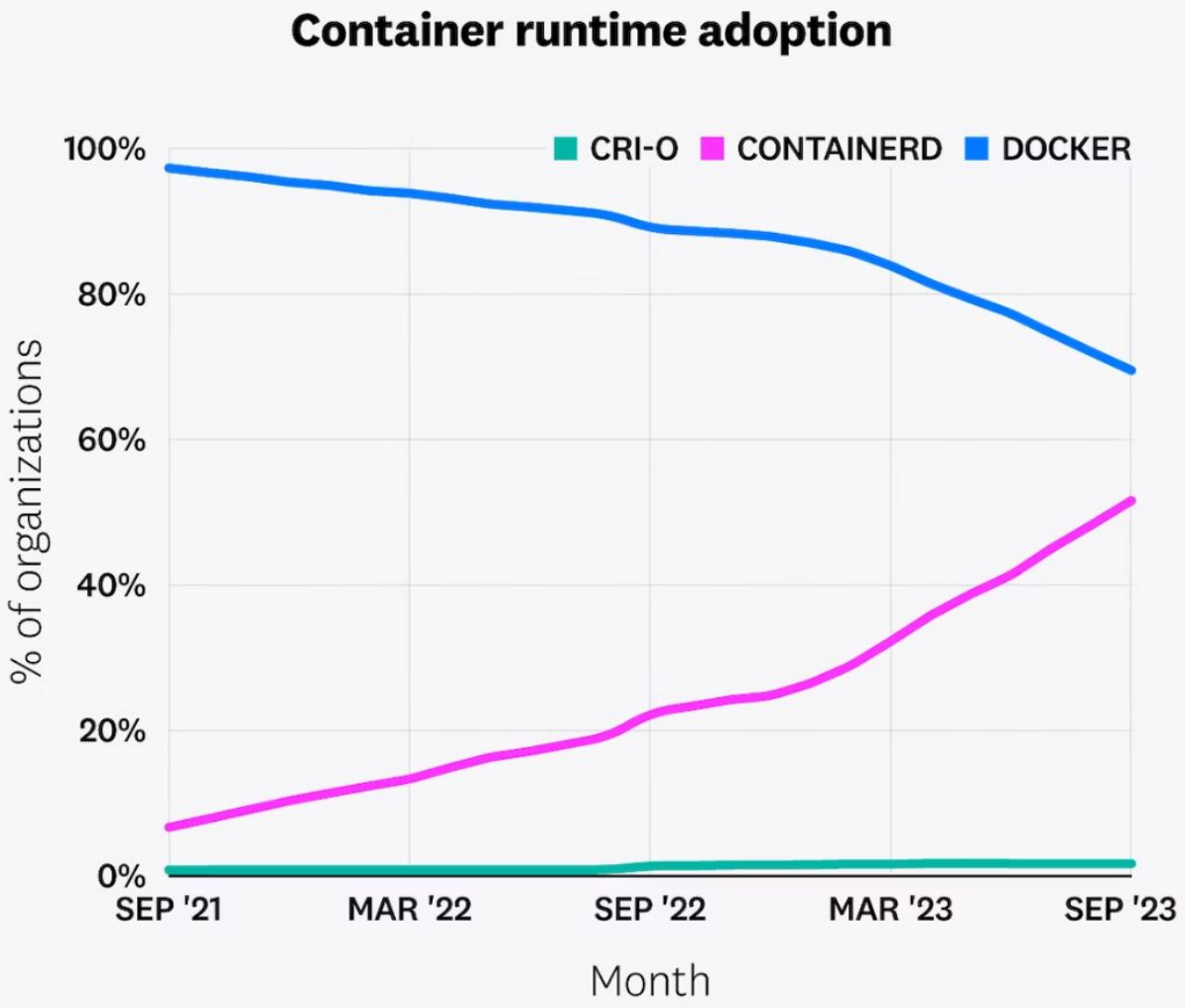

KubeCon | CloudNativeCon

North America 2023

What's new in containerd 2.0

Phil Estes, AWS
Derek McGowan, Docker

Containerd usage growth



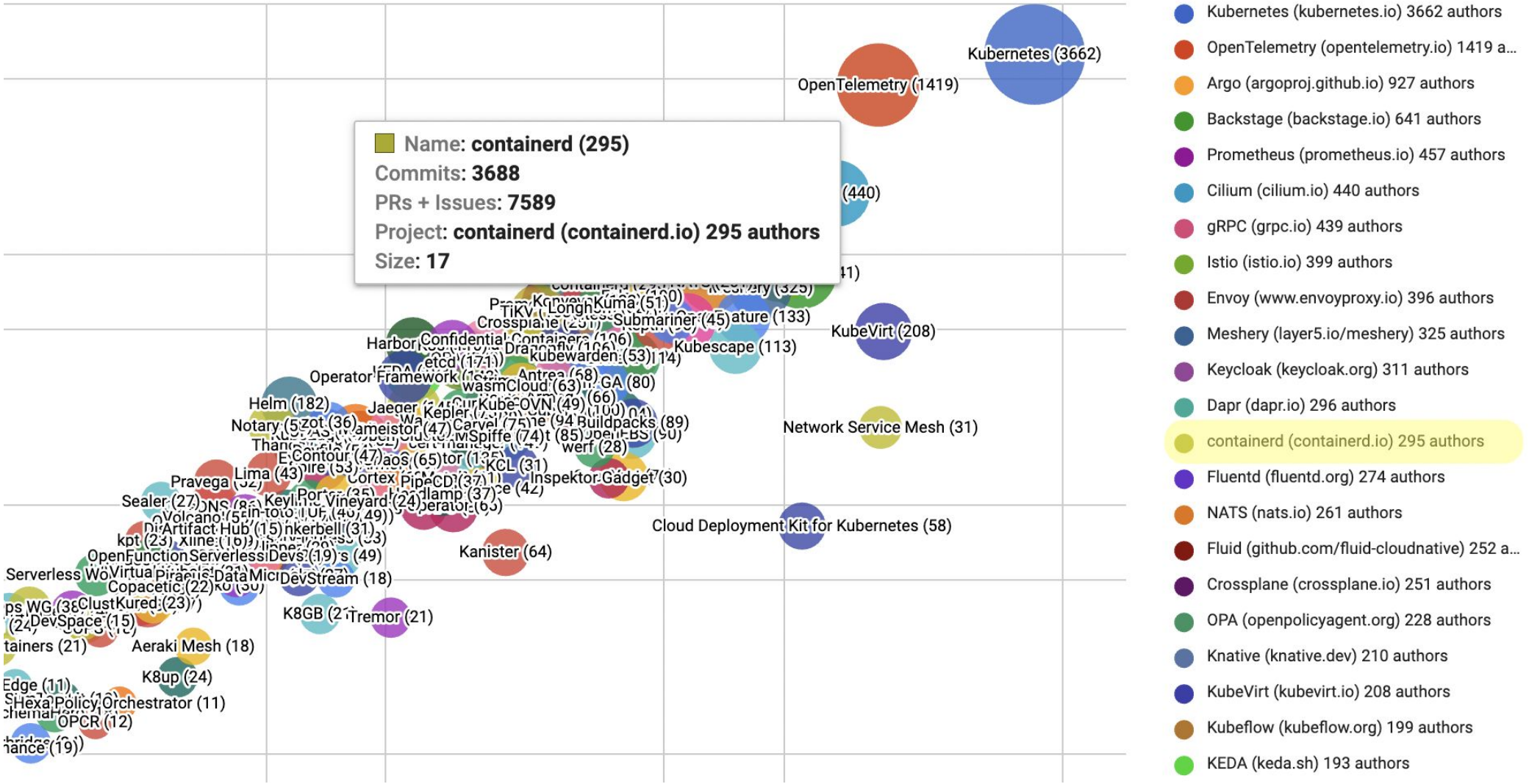
CONTAINERD ADOPTION HAS MORE THAN DOUBLED IN THE LAST YEAR

Source: Datadog

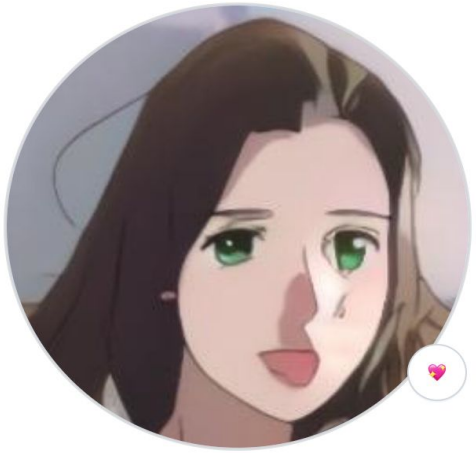
https://www.datadoghq.com/container-report/

Community growth

KubeCon CloudNativeCon North America 2023

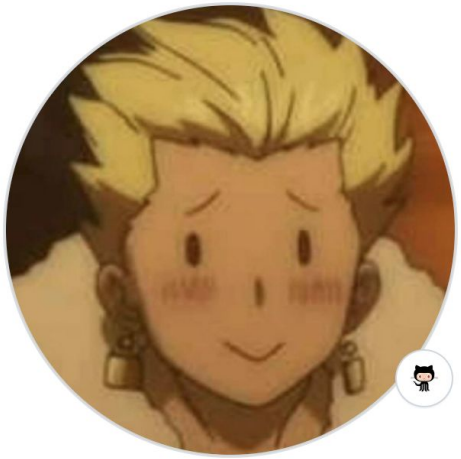


New maintainers



Laura Brehm
laurazard · she/her

kiashok · she/her



Iceber Gu
Iceber

Krisztian Litkey
klihub

Henry Wang
henry118

Ruiwen Zhao
ruiwen-zhao

Akhil Mohan
akhilerm

containerd adoption in K8s distros

- Alibaba Cloud Container Service for Kubernetes
- Amazon Elastic Kubernetes Service
- Azure Kubernetes Service
- Google Kubernetes Engine
- Huawei Cloud Cloud Container Engine
- IBM Cloud Kubernetes Service
- Rancher K3s
- VMware Tanzu
- Volcengine Kubernetes Engine

RKE2

Alibaba Group

IBM Cloud

Google Cloud

RANCHER

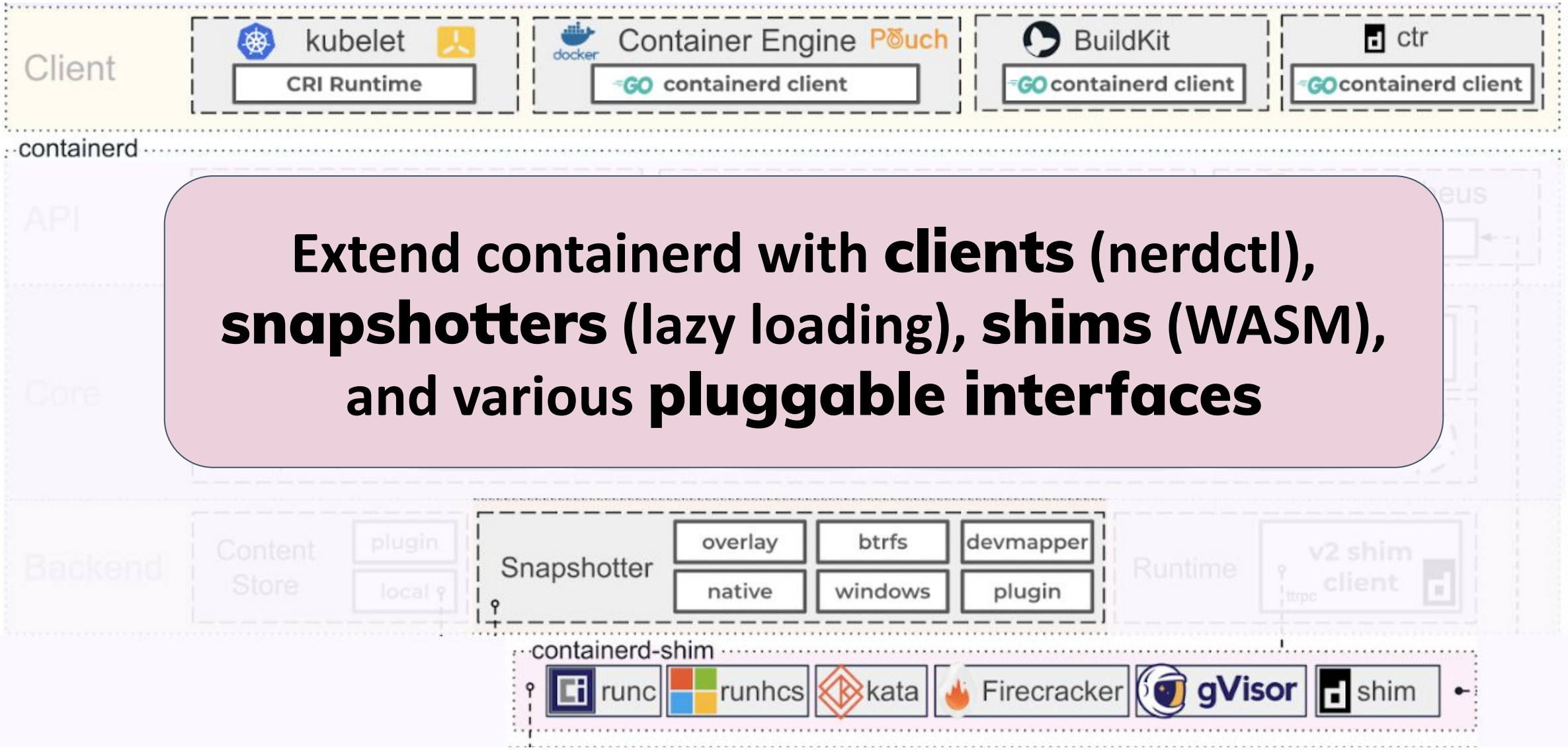
VMware Tanzu

HUAWEI CLOUD

K3S

火山引擎

Built for Extensibility



Containerd Clients

ctr	nerdctl	crictl	docker(moby)
command-line dev tool, part of core containerd project	non-core containerd project, Docker-like CLI added features for lazy-loaded, image encryption, signing	a CLI for the CRI API; a Kubernetes project (cri-tools)	further integration with containerd image store allows use of snapshotters, containerd features

- developer platforms using containerd -

- **Colima** - container runtimes on macOS (and Linux) with minimal setup
- **Finch** - Docker-like CLI for MacOS
- **Rancher Desktop** - Docker-like experience on MacOS, Windows, and Linux

Snapshotters

Extendable via proxy plugins

Built In/Core Snapshotters

- overlayfs (Linux)
- devmapper (Linux)
- btrfs (Linux)
- native (Linux/Unix/Windows)
- blockfile (New! Linux/Unix)
- zfs (Linux/Unix)
- LCOW (Windows)
- Windows (Windows)

Remote Snapshotters

- stargz (Filesystem, non-core project)
- overlaybd (Block, non-core project)
- nydus (Filesystem, non-core project)
- SOCI (Filesystem, OSS vendor project)
- GKE image streaming (Filesystem, vendor project)

Runtimes & Shims

OCI Runtimes

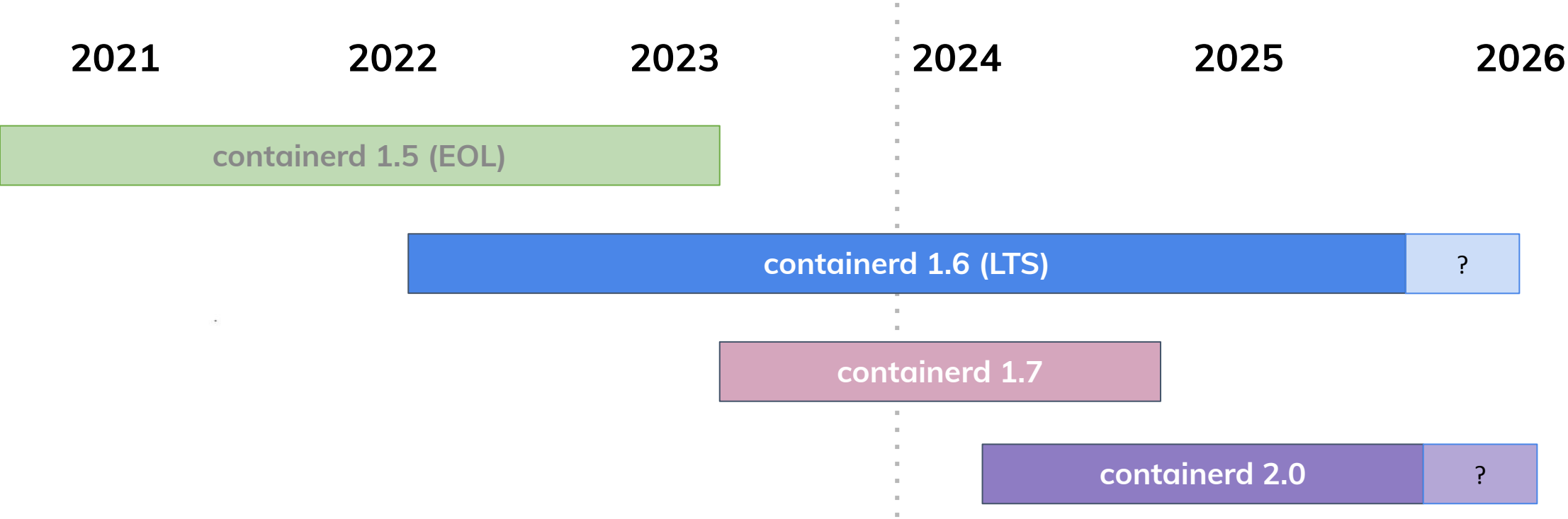
- **runc** - default Linux OCI runtime
- **crun** - alternative Linux OCI runtime in C
- **youki** - alternative Linux OCI runtime in Rust
- **runj** - experimental FreeBSD OCI runtime

External Shim Projects

- **hcsshim/runhcs** - shim and OCI runtime for Windows
- **runwasi** - (New! Non-core project) - WASM shim
- **Kata Containers** - hypervisor-based isolation for pods
- **gVisor/runsc** - independent kernel for isolation
- **firecracker-containerd** - hypervisor-based isolation for containers based on Firecracker
- **inclavare-containers** - shim for hardware-assisted Trusted Execution Environment (TEE)
- **kuasar** - shim supporting multiple sandbox techniques

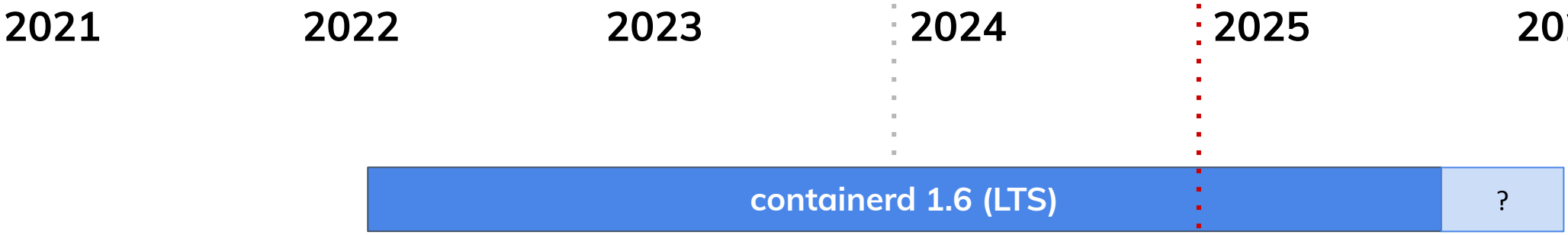
- **embedshim** - An ebpf-based container task runtime manager

Supported Releases



Release	Status	Start	End of Life
1.5	End of Life	May 3, 2021	February 28, 2023
1.6	LTS	February 15, 2022	max(February 15, 2025 or next LTS + 6 months)
1.7	Active	March 10, 2023	max(March 10, 2024 or release of 2.0 + 6 months)
2.0	Next	TBD	TBD

containerd v1.6 - first LTS!

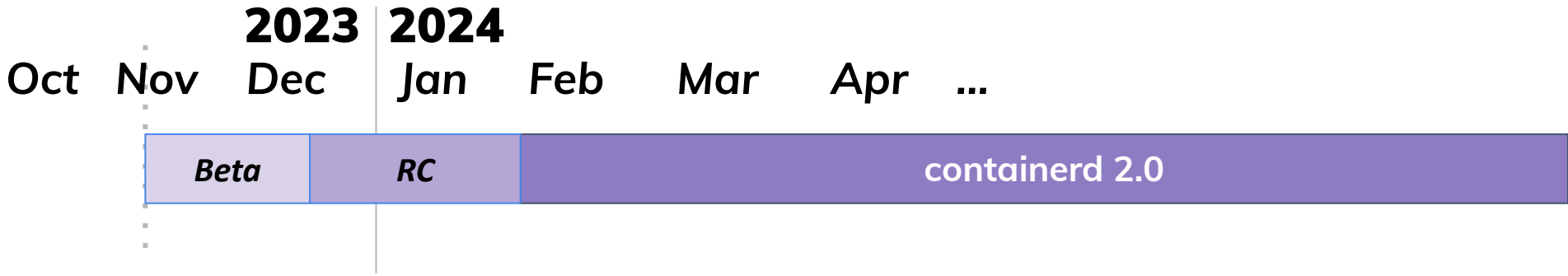


- Supported at least until **Feb 2025**
- Project has **expanded scope** for backports to LTS line:
  - updating library dependencies
  - stay with supported toolchain versions (Go releases)
  - keeping compatibility with current Kubernetes versions
- Will move to Active release with stricter backport criteria ~ **Aug 2024**

containerd v1.7 - last 1.x release

- **Sandbox Service and API** (New! - Experimental)
  - Shim-level API to support groups of containers
  - Preview CRI Plugin v2 - *ENABLE_CRI_SANDBOXES=1*
- **Node Resource Interface** (Updated - Experimental)
  - Extensions for OCI-compatible container runtimes
  - TTRPC
- **Transfer Service** (New! - Experimental)
  - Support to transfer artifact objects between any source and destination
- **CRI User-Namespace Support** (New! - Experimental)
- **gRPC Shim Support** (New! - Experimental)

v2.0 - Release Plan



Component	Initial Release	Target Supported Release
Sandbox Service	containerd v1.7	containerd v2.0
Sandbox CRI Server	containerd v1.7	containerd v2.0
Transfer Service	containerd v1.7	containerd v2.0
NRI in CRI Support	containerd v1.7	containerd v2.0
gRPC Shim	containerd v1.7	containerd v2.0
CRI Runtime Specific Snapshotter	containerd v1.7	containerd v2.0
CRI Support for User Namespaces	containerd v1.7	containerd v2.0

v2.0 - Removed Features

Component	Deprecation release	Target release for removal	Recommendation
Runtime V1 API and implementation (`io.containerd.runtime.v1.linux`)	containerd v1.4	containerd v2.0 ✅	Use `io.containerd.runc.v2`
Runc V1 implementation of Runtime V2 (`io.containerd.runc.v1`)	containerd v1.4	containerd v2.0 ✅	Use `io.containerd.runc.v2`
config.toml `version = 1`	containerd v1.5	containerd v2.0 ✅	Use config.toml `version = 2`
Built-in `aufs` snapshotter	containerd v1.5	containerd v2.0 ✅	Use `overlayfs` snapshotter
Container label `containerd.io/restart.logpath`	containerd v1.5	containerd v2.0 ✅	Use `containerd.io/restart.loguri` label
`cri-containerd-*.tar.gz` release bundles	containerd v1.6	containerd v2.0 ✅	Use `containerd-*.tar.gz` bundles
Pulling Schema 1 images (`application/vnd.docker.distribution.manifest.v1+json`)	containerd v1.7	containerd v2.0	Use Schema 2 or OCI images
CRI `v1alpha2`	containerd v1.7	containerd v2.0 ✅	Use CRI `v1`
Legacy CRI implementation of podsandbox support	containerd v2.0	containerd v2.1	Disabled by default in 2.0 in favor of core sandboxed CRI plugin (use `DISABLE_CRI_SANDBOXES=1` to fallback to prior CRI podsandbox implementation)

v2.0 - Config Migration

Property Group	Property	Deprecation release	Target release for removal	Recommendation
`[plugins."io.containerd.grpc.v1.cri"]`	`systemd_c`	erd	containerd v2.0 ✅	Use `SystemdCgroup` in runc options (see below)
`[plugins."io.containerd.grpc.v1.cri".con`			containerd v2.0 ✅	Create `untrusted` runtime in `runtimes`
`[plugins."io.containerd.grpc.v1.cri".containerd`		ntainerd v1.3	containerd v2.0 ✅	Use `default_runtime_name`
`[plugins."io.containerd.grpc.v1.cri".containerd.runtime`		containerd v1.3	containerd v2.0 ✅	Use runtime v2
`[plugins."io.containerd.grpc.v1.cri".containerd.r`		containerd v1.3	containerd v2.0 ✅	Use `options.Root`
`[plugins."io.containerd.grpc.v1.cri".conta`		inerd	containerd v2.0 ✅	Set `$PATH` to the `criu` binary
`[plugins."io.containerd.grpc.v1.cri".re`	`auths`		containerd v2.0	Use `ImagePullSecrets`. See also #8228.
`[plugins."io.containerd.grpc.v1.cri".registry]`	`configs`	ntainerd v1.5	containerd v2.0	Use `config_path`
`[plugins."io.containerd.grpc.v1.cri".registry]`	`mirrors`	containerd v1.5	containerd v2.0	Use `config_path`

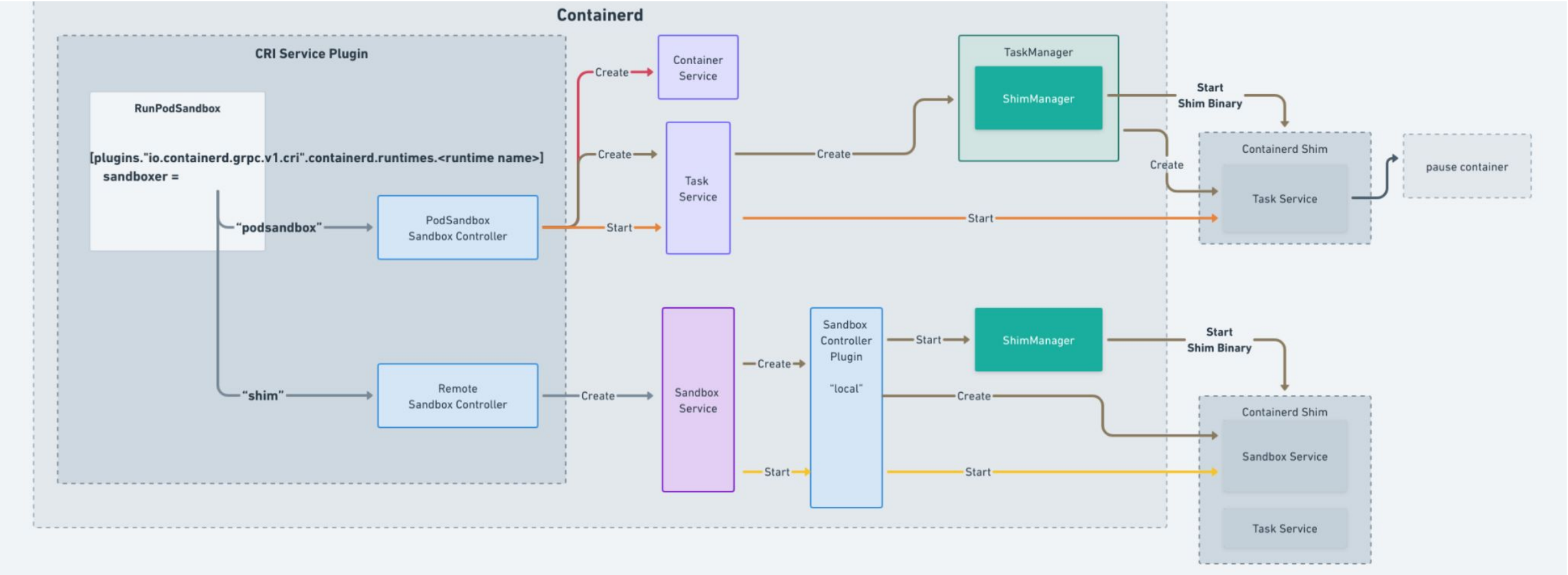
Instead of **deprecating** the config v1 properties and format, we introduced **config migration** which converts to the new config format and properties on the users behalf in v2.0

Sandbox API

- New Sandbox API to group containers for shim
  - Separates sandbox lifecycle from container lifecycle
- Sandbox Controller interface
  - Handle sandbox environment for grouped containers
  - Support to manage multiple runtime platforms
    - Linux/Unix/Windows
    - Container, VM, microVM
- Controllers registered as containerd plugins
  - “sandboxer” configuration used to choose between sandbox controllers, similar to choosing between snapshotters.

```
service Sandbox {
        // CreateSandbox will be called right after sandbox shim instance launched.
        // It is a good place to initialize sandbox environment.
        rpc CreateSandbox(CreateSandboxRequest) returns (CreateSandboxResponse);

        // StartSandbox will start a previously created sandbox.
        rpc StartSandbox(StartSandboxRequest) returns (StartSandboxResponse);

        // Platform queries the platform the sandbox is going to run containers on.
        // containerd will use this to generate a proper OCI spec.
        rpc Platform(PlatformRequest) returns (PlatformResponse);

        // StopSandbox will stop existing sandbox instance
        rpc StopSandbox(StopSandboxRequest) returns (StopSandboxResponse);

        // WaitSandbox blocks until sandbox exits.
        rpc WaitSandbox(WaitSandboxRequest) returns (WaitSandboxResponse);

        // SandboxStatus will return current status of the running sandbox instance
        rpc SandboxStatus(SandboxStatusRequest) returns (SandboxStatusResponse);

        // PingSandbox is a lightweight API call to check whether sandbox alive.
        rpc PingSandbox(PingRequest) returns (PingResponse);

        // ShutdownSandbox must shutdown shim instance.
        rpc ShutdownSandbox(ShutdownSandboxRequest) returns (ShutdownSandboxResponse);

        // SandboxMetrics retrieves metrics about a sandbox instance.
        rpc SandboxMetrics(SandboxMetricsRequest) returns (SandboxMetricsResponse);
}
```

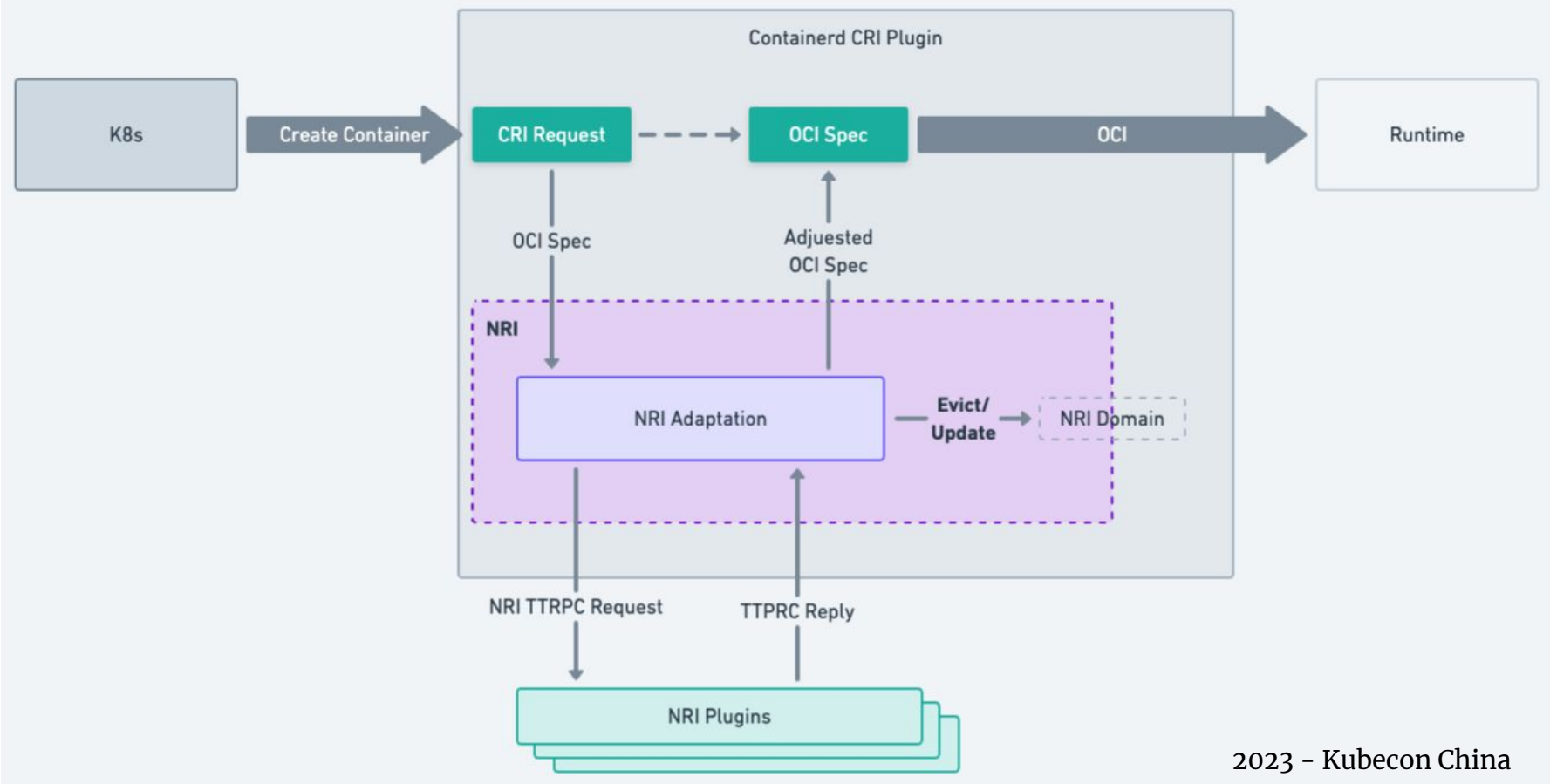
Sandbox API：Controller & Service



Node Resource Interface

- Middleware extension between CRI and OCI
- ttrpc bindings

Create a Container

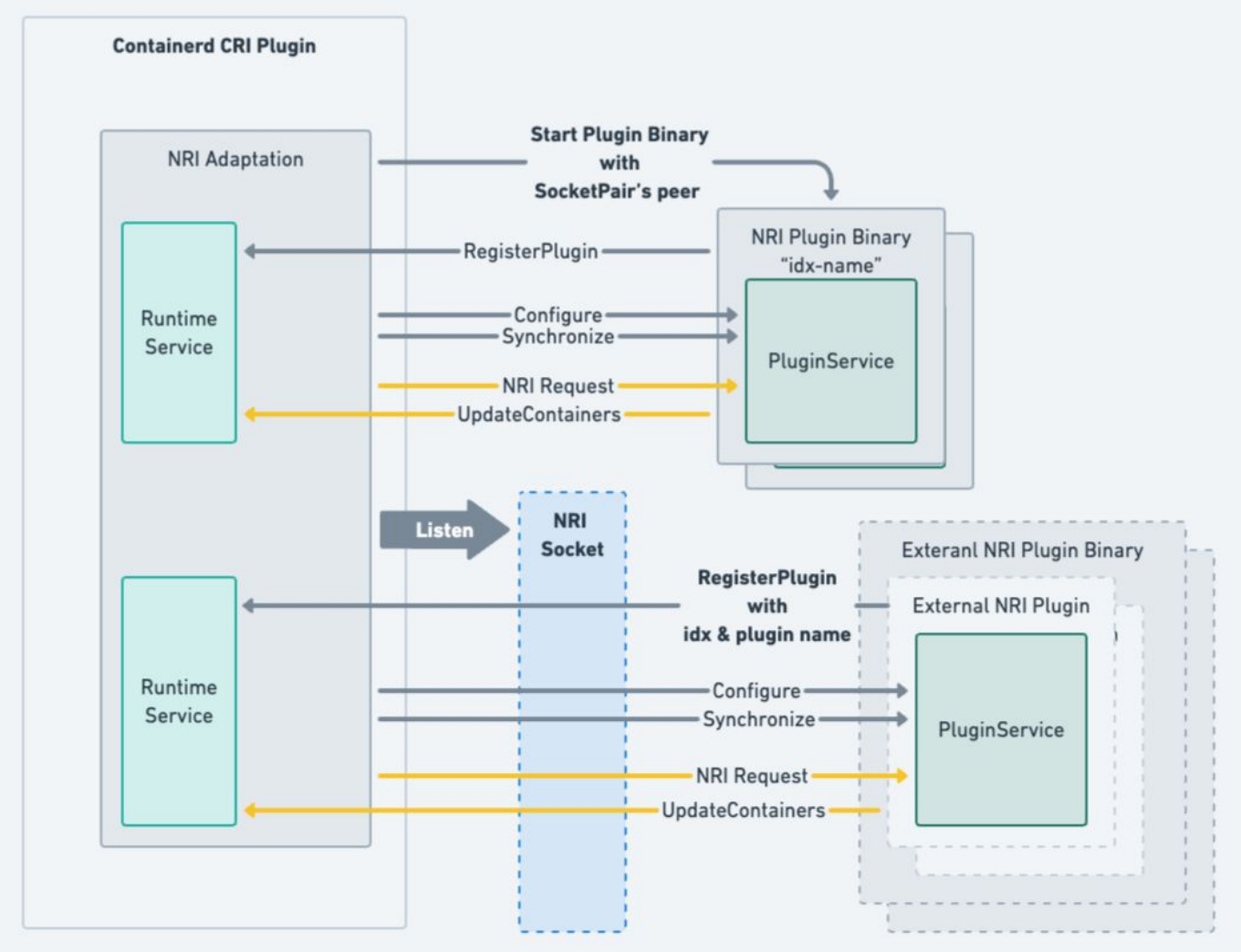


Node Resource Interface

Plugin Registration

- NRI Plugin Binary
- External NRI Plugin

```
[plugins."io.containerd.nri.v1.nri"]
    # Enable NRI support in containerd.
    disable = false

    # Allow connections from externally launched NRI plugins.
    disable_connections = false

    # plugin_config_path is the directory to search for plugin-specific configuration.
    plugin_config_path = "/etc/nri/conf.d"

    # plugin_path is the directory to search for plugins to launch on startup.
    plugin_path = "/opt/nri/plugins"

    # plugin_registration_timeout is the timeout for a plugin to register after connection.
    plugin_registration_timeout = "5s"

    # plugin_requst_timeout is the timeout for a plugin to handle an event/request.
    plugin_request_timeout = "2s"

    # socket_path is the path of the NRI socket to create for plugins to connect to.
    socket_path = "/var/run/nri/nri.sock"
```



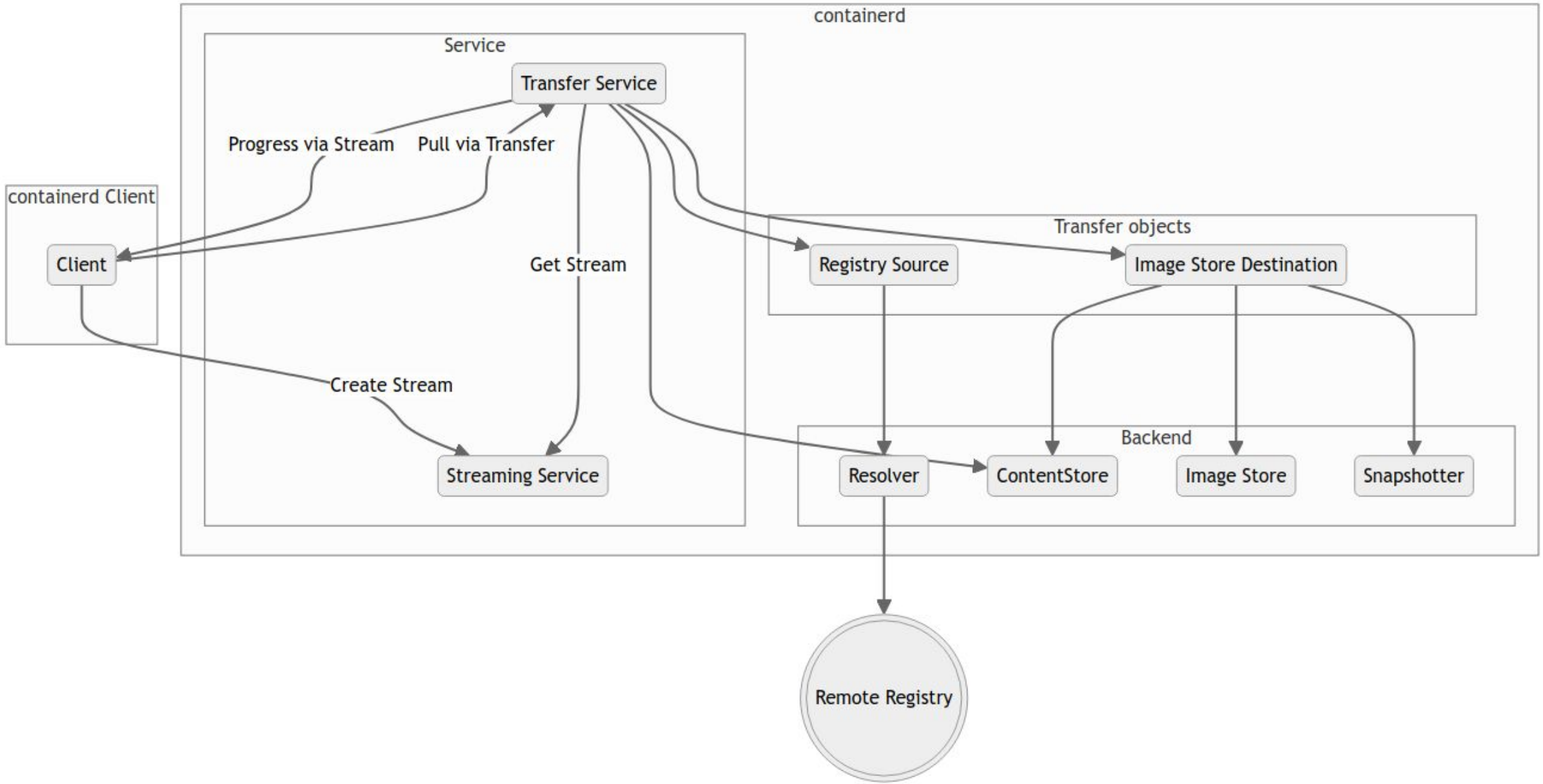
Transfer Service

Simple interface

```
Transfer(ctx context.Context, source interface{}, destination interface{}, opts ...Opt) error
```

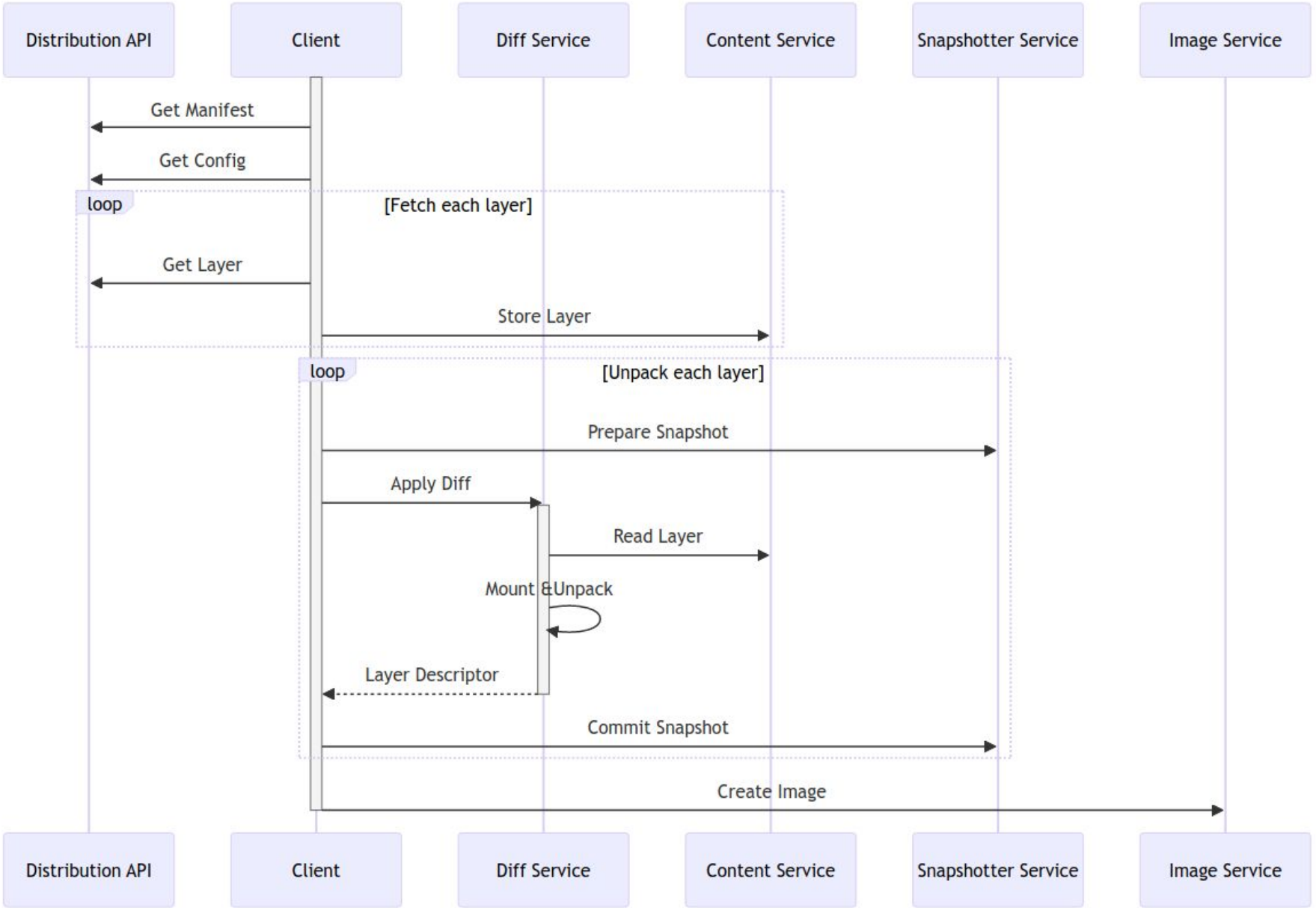
Source	Destination	Description
Registry	Image Store	"pull"
Image Store	Registry	"push"
Object stream (Archive)	Image Store	"import"
Image Store	Object stream (Archive)	"export"
Object stream (Layer)	Mount/Snapshot	"unpack"
Mount/Snapshot	Object stream (Layer)	"diff"
Image Store	Image Store	"tag"
Registry	Registry	mirror registry image

Transfer Service



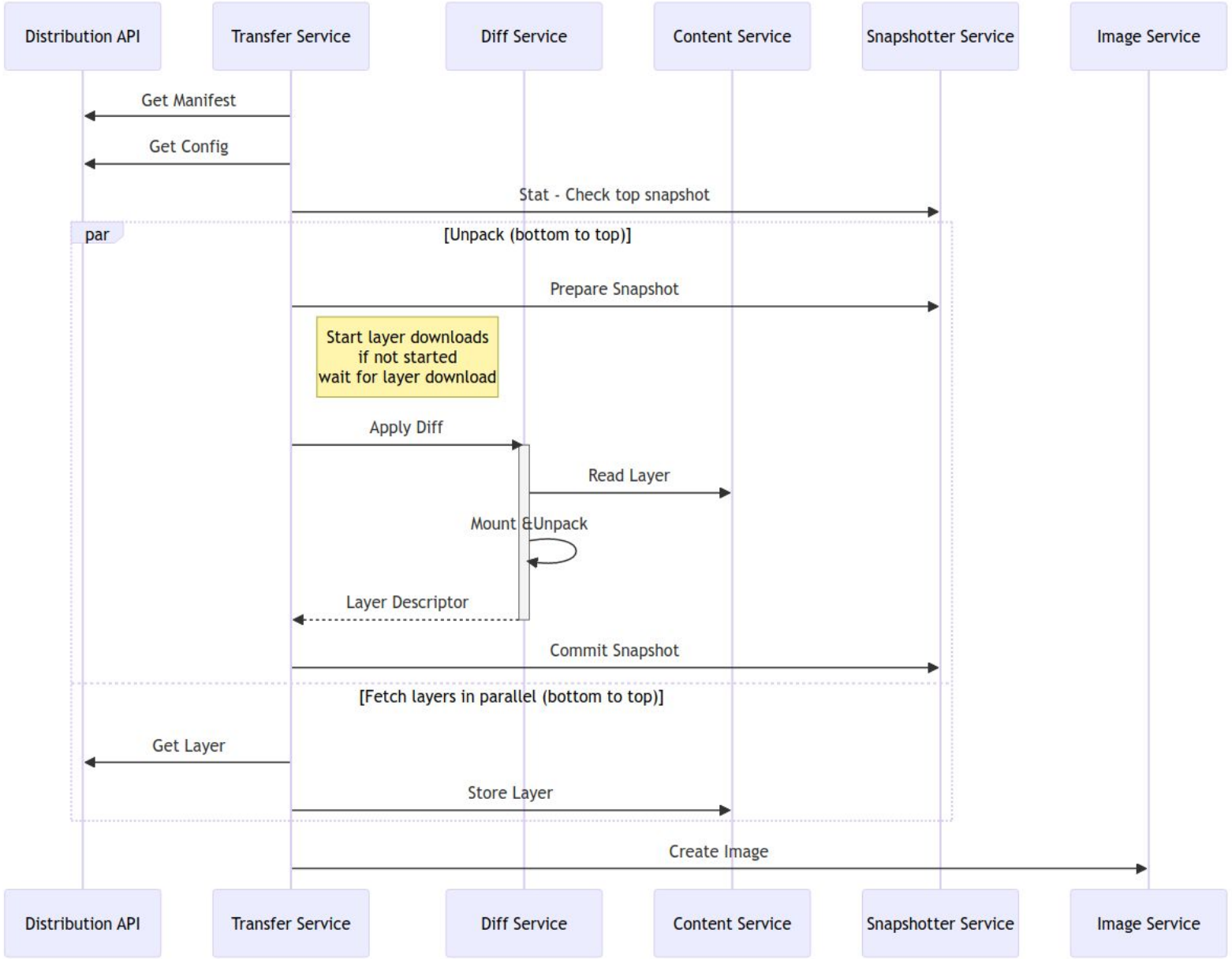
Transfer Service

"Simple" Pull



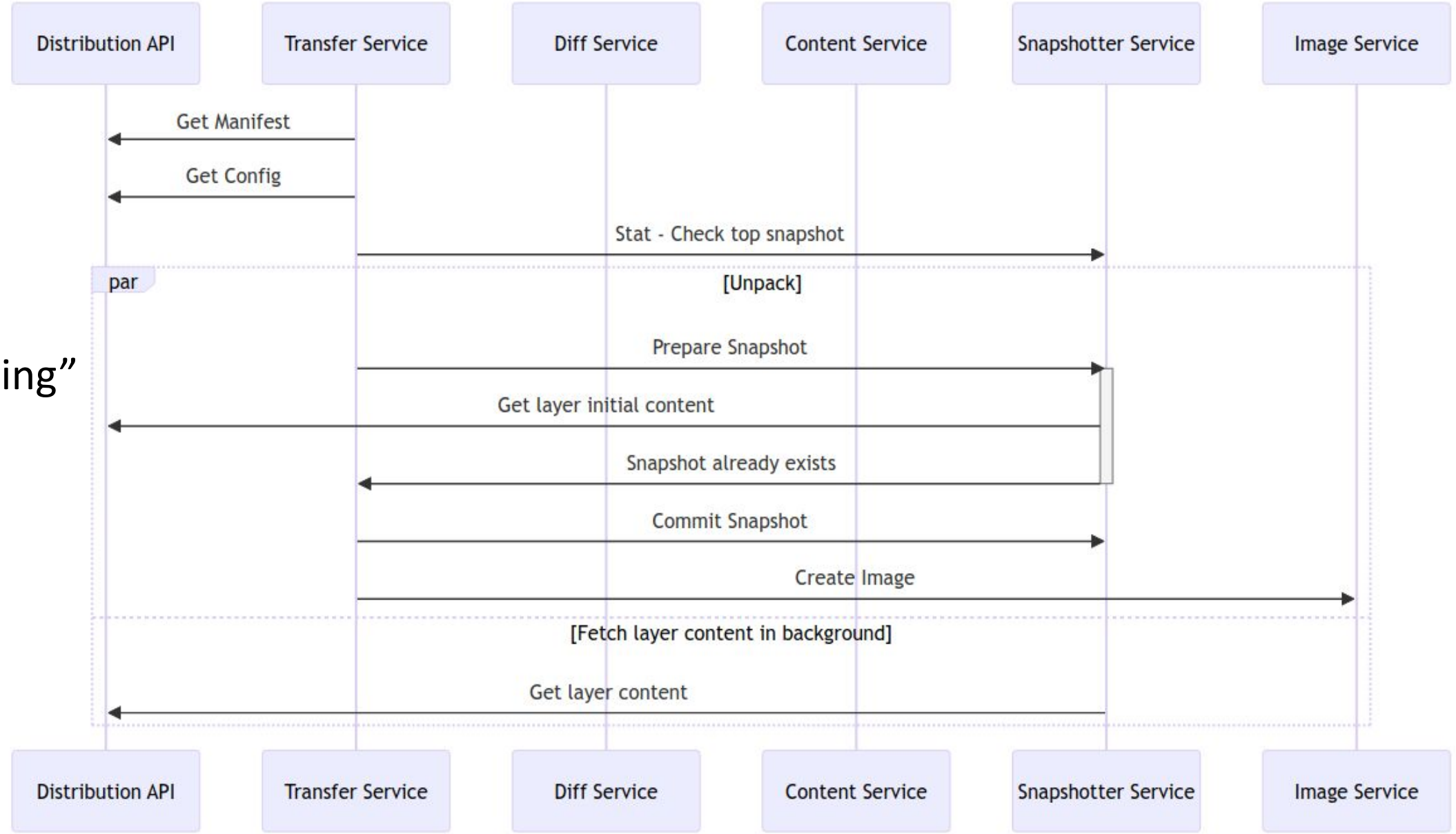
Transfer Service

With
parallel unpack



Transfer Service

With
“lazy loading”



Transfer Service

- New use-cases and extensions
  - Confidential computing (guest sandbox env is the destination)
  - OCI Referrers API support (mountable images, lazy-loading images)
  - Signing and image validation
    - [transfer] plugin to transfer service for image verification
    - Support Ratify as a containerd plugin
  - Plugins to customize image pulling logic
  - Credential Management
- Enable Transfer Service in CRI plugin by default

User-Namespace Support in CRI

- **Support for user namespaces in stateless pods (v1.7)**
  - Only support emptyDir, configmap, secret, downwardsAPI
  - Use chown and cache the snapshots with same mapping
- **Supports Running Stateful Pods in (v2.0)**
  - Integrated with Idmapped mount
  - User Namespaces: Now Supports Running Stateful Pods in Alpha

Future & In-Development

- **Transfer Service in CRI (in development)**
  - Credential Management
  - Image Validation
- **Shim Plugins (in development)**
- **Higher level image service**
  - Replace CRI image cache
  - Simplify clients such as nerdctl and Moby
- **Higher level container service**
  - Simpler interface for starting containers
  - Use of streaming service for IO through the API

Getting involved

- #containerd and #containerd-dev channel on
  - CNCF Slack (https://slack.cncf.io)
- **Community Meeting on the 2nd and 4th Thursday of every month**
  - See CNCF Calendar for your timezone (https://cncf.io/calendar)
- Build something in the ecosystem!
- Discussion, issues and pull requests welcome!
  - https://github.com/containerd/containerd