Dockyard-container

Image Registry & Volume Management

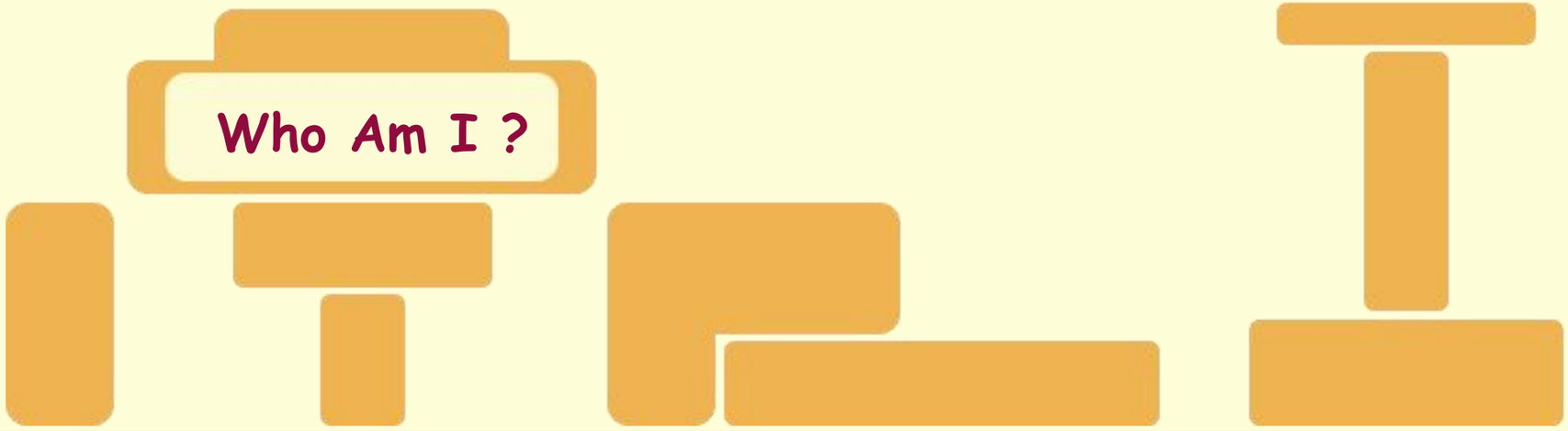
https://github.com/containerops/dockyard

Leo Meng <mengfanlaing@huawei.com>

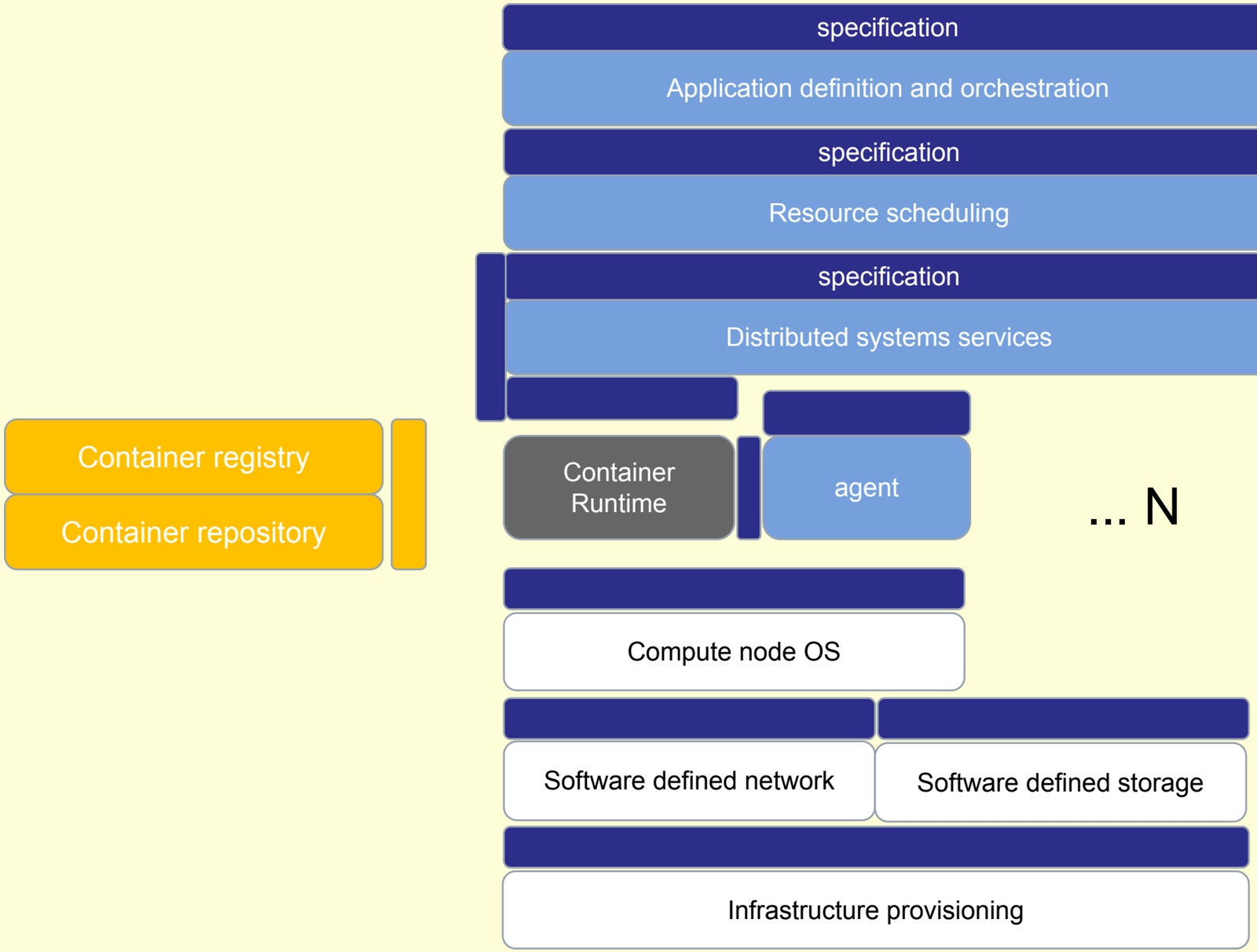
孟繁亮

Senior Architect & Full Stack Developer

Email: mengfanliang@huawei.com

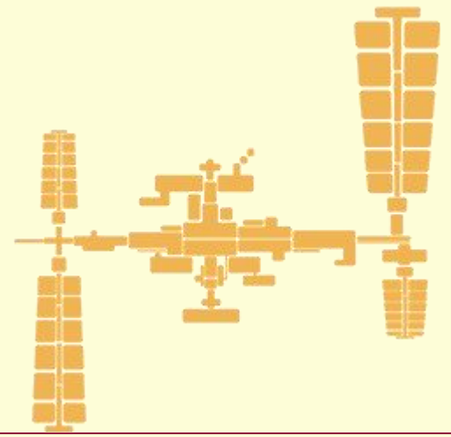


CNCF Conceptual Architecture

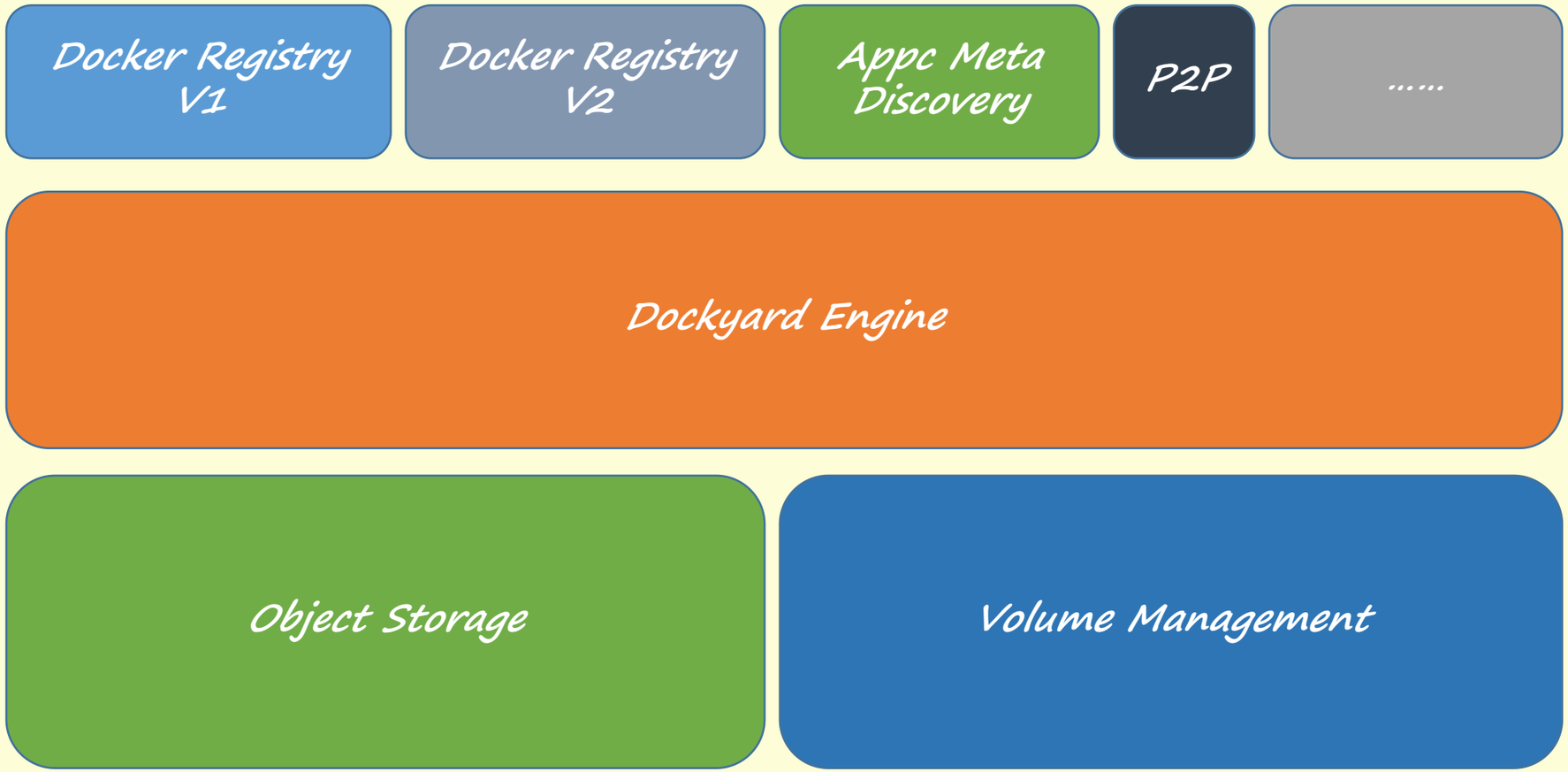


Dockyard Proposals

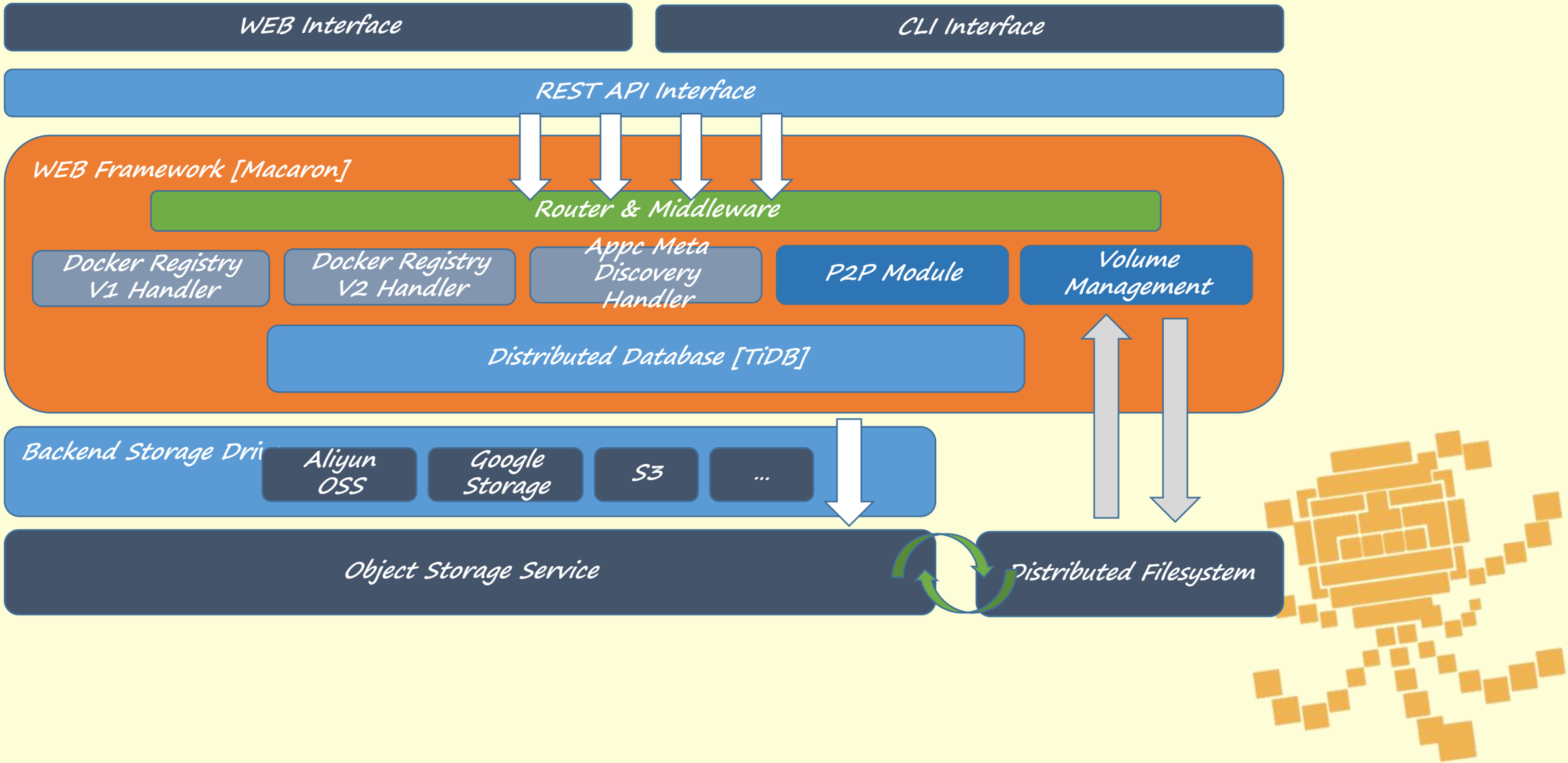
- Image storage and meta discovery for major container runtime like Docker, rkt, hyper.
- Support P2P delivery image.
- Public & private repositories for user and organization.
- Build-in object storage engine and drivers.
- Build-in object storage service.
- Build-in container volume management for runtime with distributed file system.
- Convert image object to distributed file-system for container mount and start directly.
- Container image encryption and verification.



Dockyard Functions



Dockyard Architecture

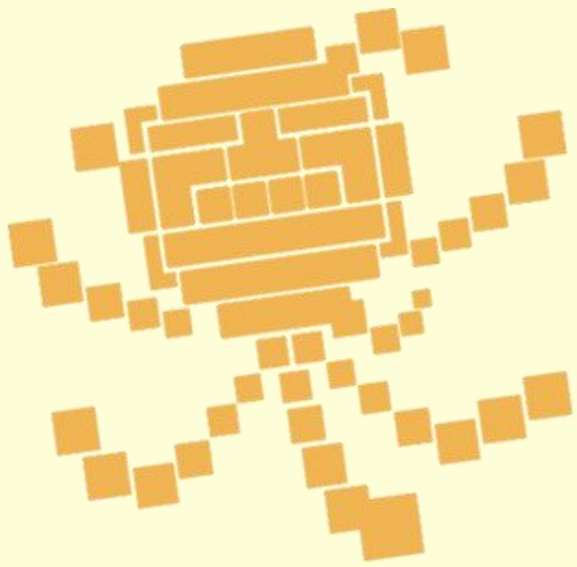


Docker Registry V1 & V2 Protocol

	Docker Registry V1	Docker Registry V2
Implement	Docker Registry	Docker Distribution
Language	Python	Golang
PULL/PUSH Resumable	None	Spec
Authentication	Authentication With Index(Hub)	Authentication Service
Index	Image ID	SHA1
Manifest	None manifest, json/layer/checksum file per layer	Have manifest with V1 compatibility
Signature	None	Signed Manifest
Method	POST PUT GET	HEAD POST PATCH PUT GET

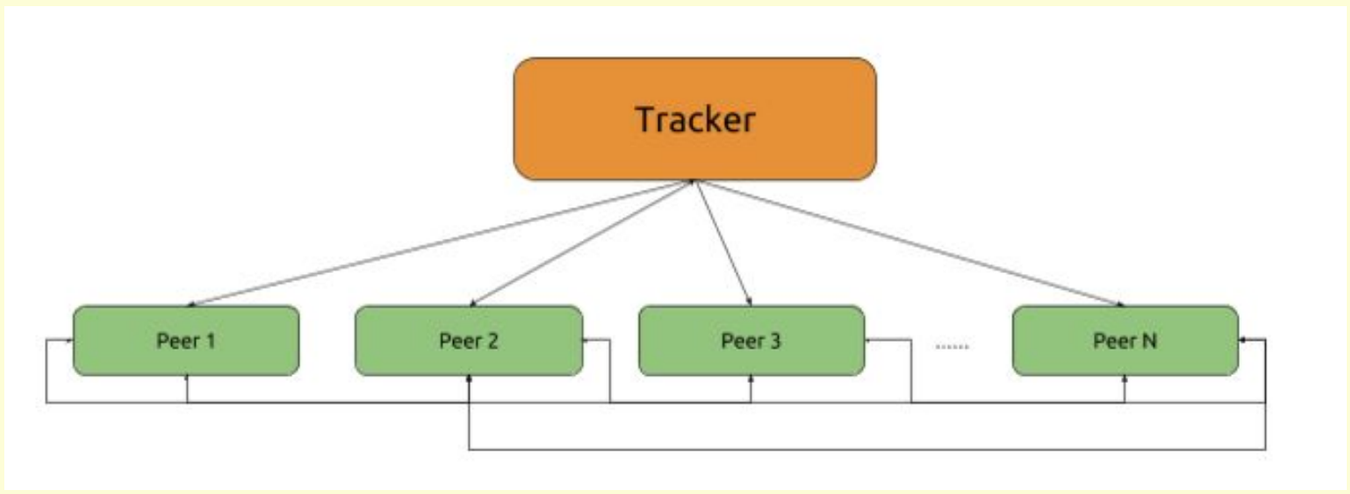
Appc Metadata Discovery Protocol

- Simple Discovery : https://{name}-{version}-{os}-{arch}.{ext}
- Meta Discovery: https://{name}?ac-discovery=1
- Validation: sha512
- Authentication: HTTP basic authentication over HTTPS connections



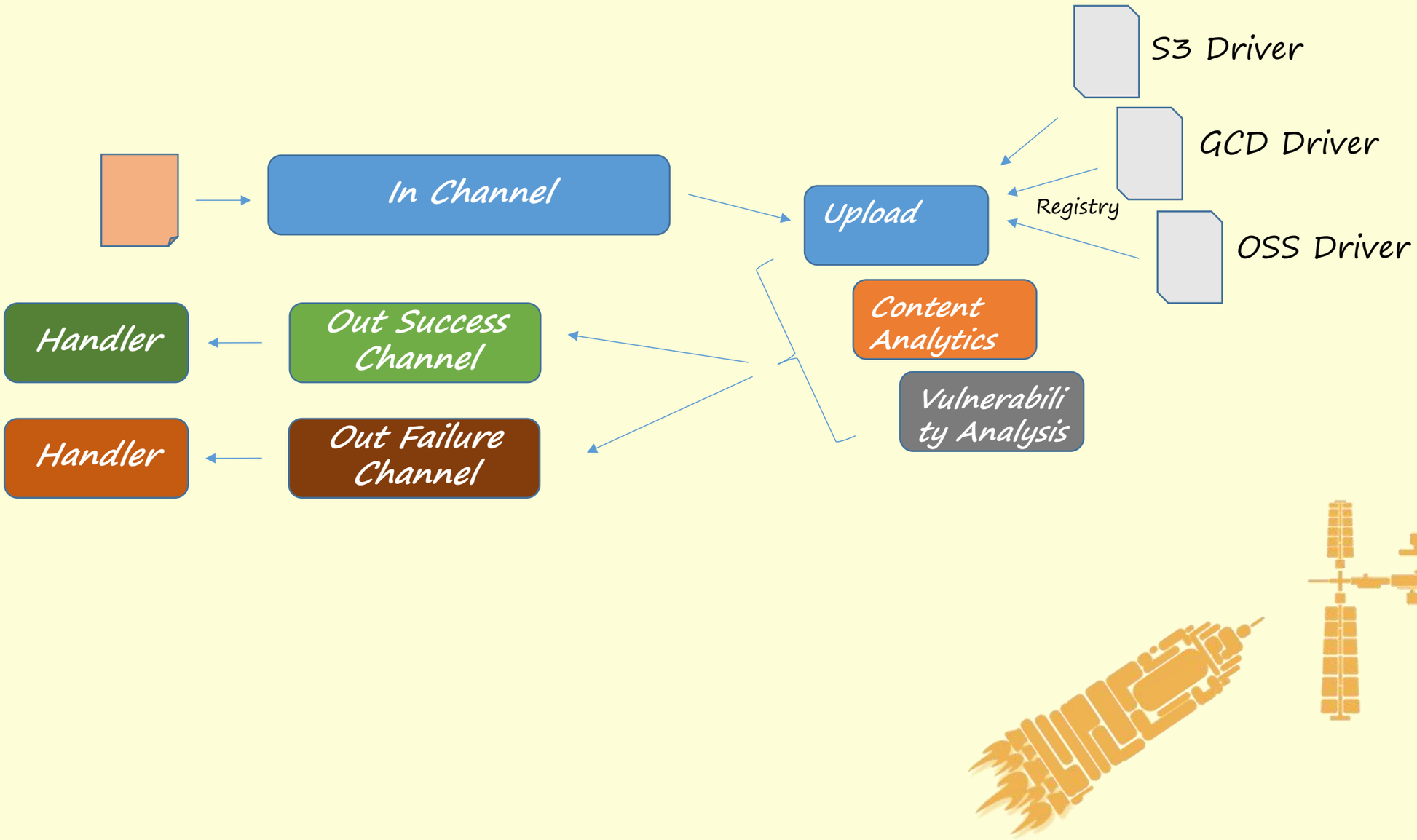
Distribute Container Image Peer To Peer

https://github.com/coreos/rkt/issues/1751



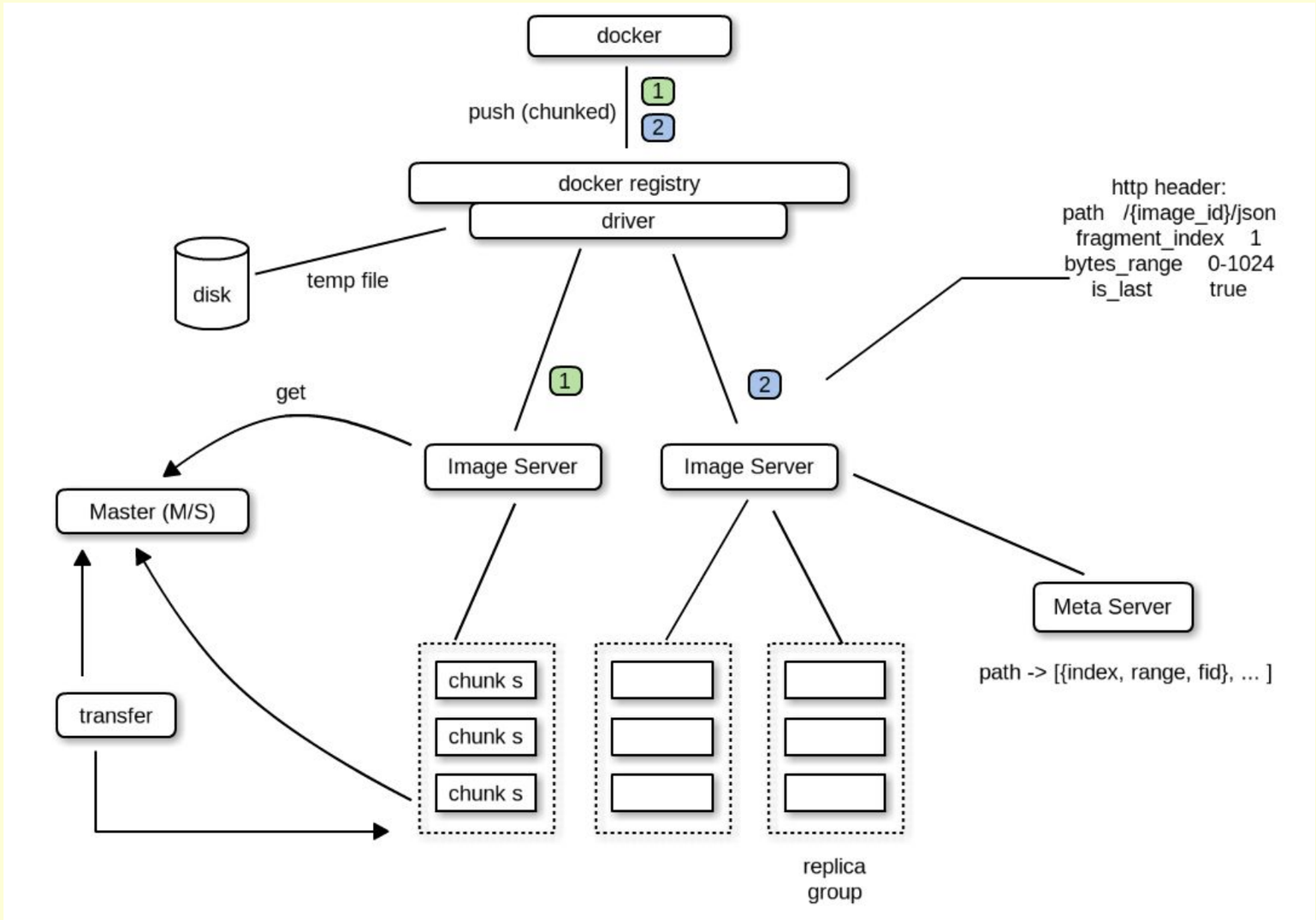

```
func runFetch(cmd *cobra.Command, args []string) (exit int) {
    if flagP2p {
        if len(args) < 1 {
            stderr("fetch: must provide tottent file.")
            return 1
        }

        tmpDir, err := ioutil.TempDir("", "rktp2pimage")
        if err != nil {
            fmt.Printf("error create temp directory: %v", err)
            return 1
        }
        defer os.RemoveAll(tmpDir)

        flags := &torrent.TorrentFlags{
            Dial:               nil,    //not use proxy
            Port:               7777,   //default port 7777
            FileDir:            tmpDir, //file to store
            SeedRatio:          math.Inf(0),
            UseDeadlockDetector: false,
            UseLPD:             false,
            UseDHT:             false,
            UseUPnP:            false,
            UseNATPMP:          false,
            TrackerlessMode:    false,
            // IP address of gateway
            Gateway:            "",
            InitialCheck:       true,
            FileSystemProvider: torrent.OsFsProvider{},
            Cacher:             nil,
            ExecOnSeeding:      "",
            QuickResume:        false,
            MaxActive:          10,
        }

        metaInfo, err := torrent.GetMetaInfo(nil, args[0])
        if err != nil {
            fmt.Printf("error parse torrent file: %v", err)
            return 1
        }
```

Storage Engine With Drivers

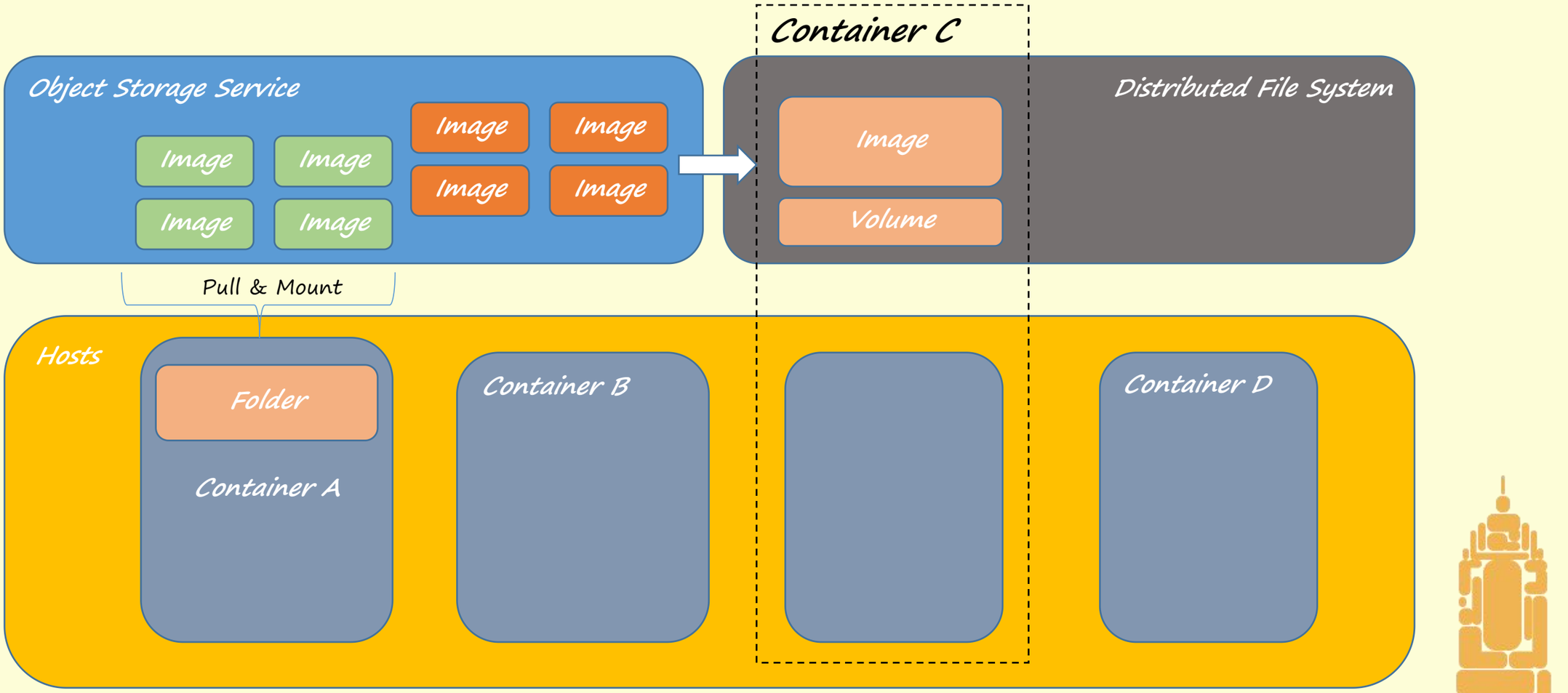


Build-in Object Storage Service

https://github.com/brewcoolbear/ipa

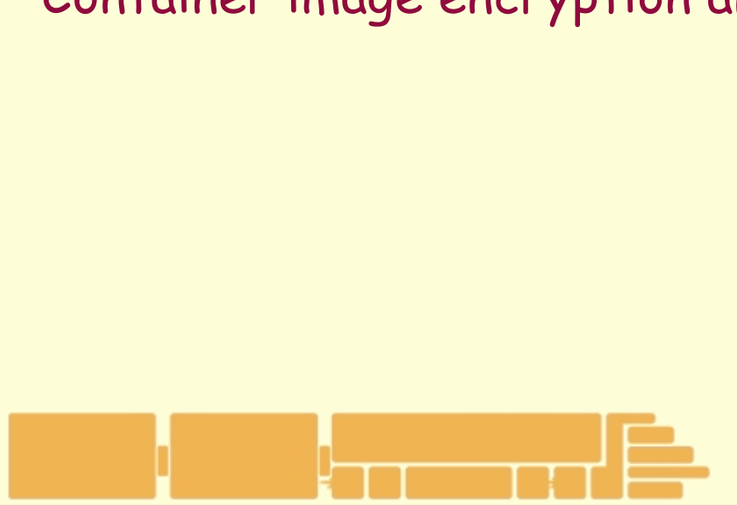


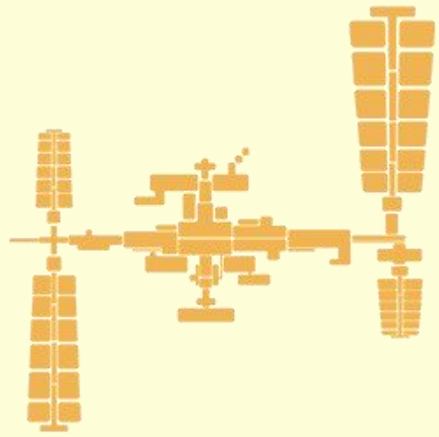
Volume Management



Dockyard Roadmap

- Docker registry V1 [Done]
- Docker registry V2 [Done]
- Object storage backend and drivers [Done]
- REST API interface [Done]
- Rkt Meta Discovery [Done]
- P2P modules [Doing]
- Object storage service [Doing]
- Volume management with distributed file-system [Feature]
- Convert between object storage and distributed file-system [Feature]
- Web interface and CLI interface [Feature]
- Container image encryption and verification[Feature]





End & Thanks