Shasta System Management Overview

CUG 2019

Harold Longley

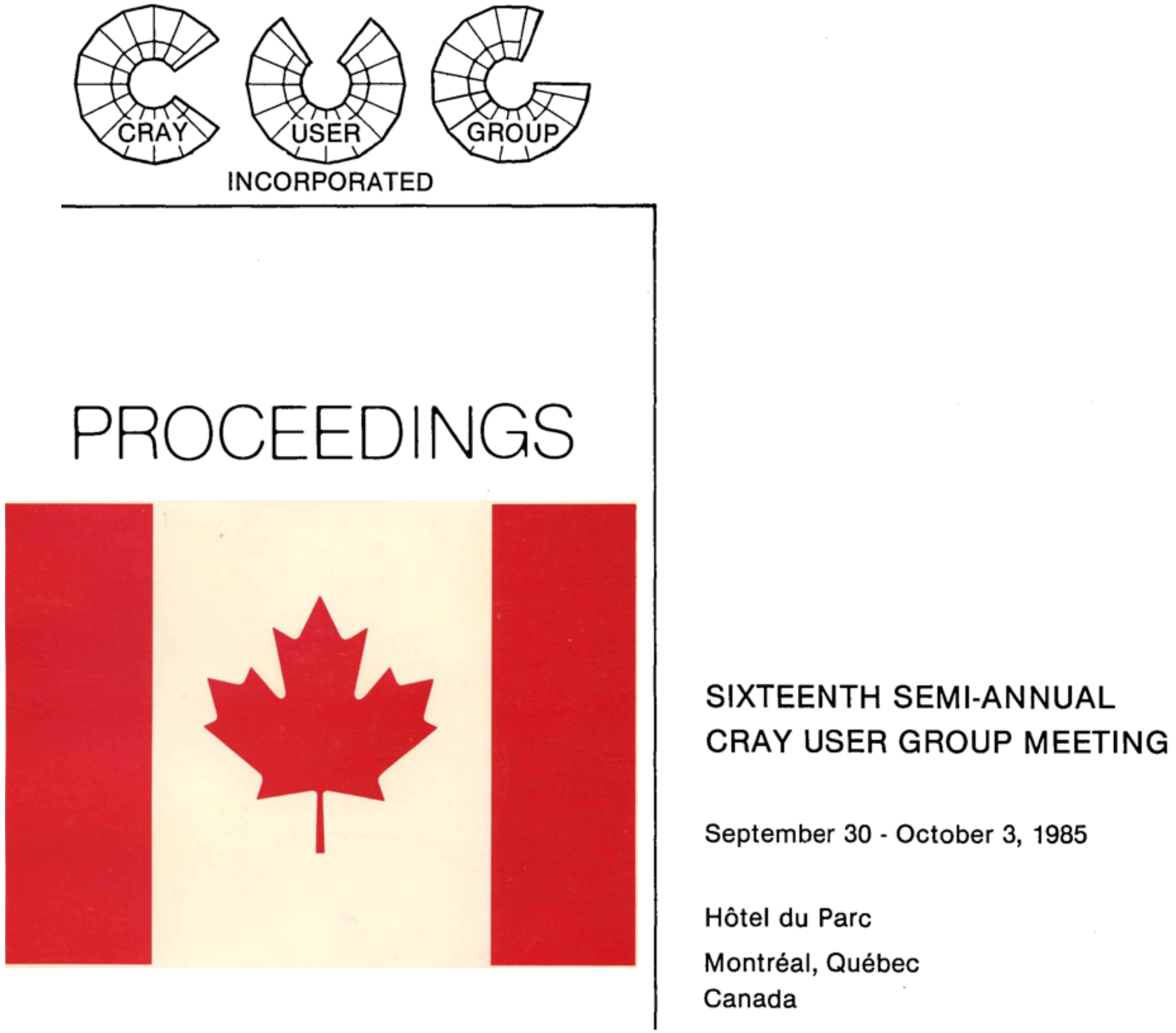
✉ htg@cray.com

Agenda – Shasta System Management Overview

- Perspective
- New hardware, new software
- Microservices with REST APIs
- Management functionality
- Shasta Presentations at CUG

Perspective

CUG in Montreal



CRAY USER GROUP

INCORPORATED

PROCEEDINGS

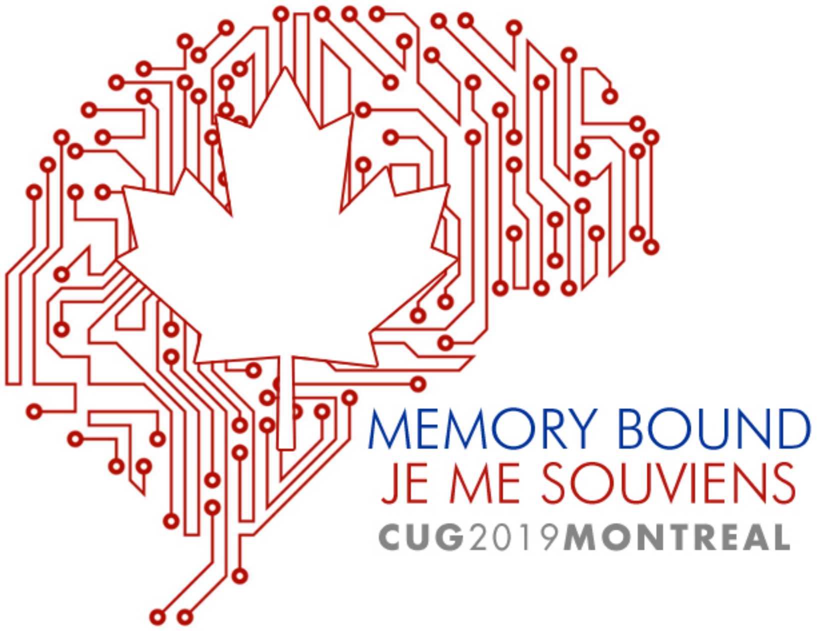
SIXTEENTH SEMI-ANNUAL
CRAY USER GROUP MEETING

September 30 - October 3, 1985

Hôtel du Parc

Montréal, Québec
Canada

CUG
Cray User Group



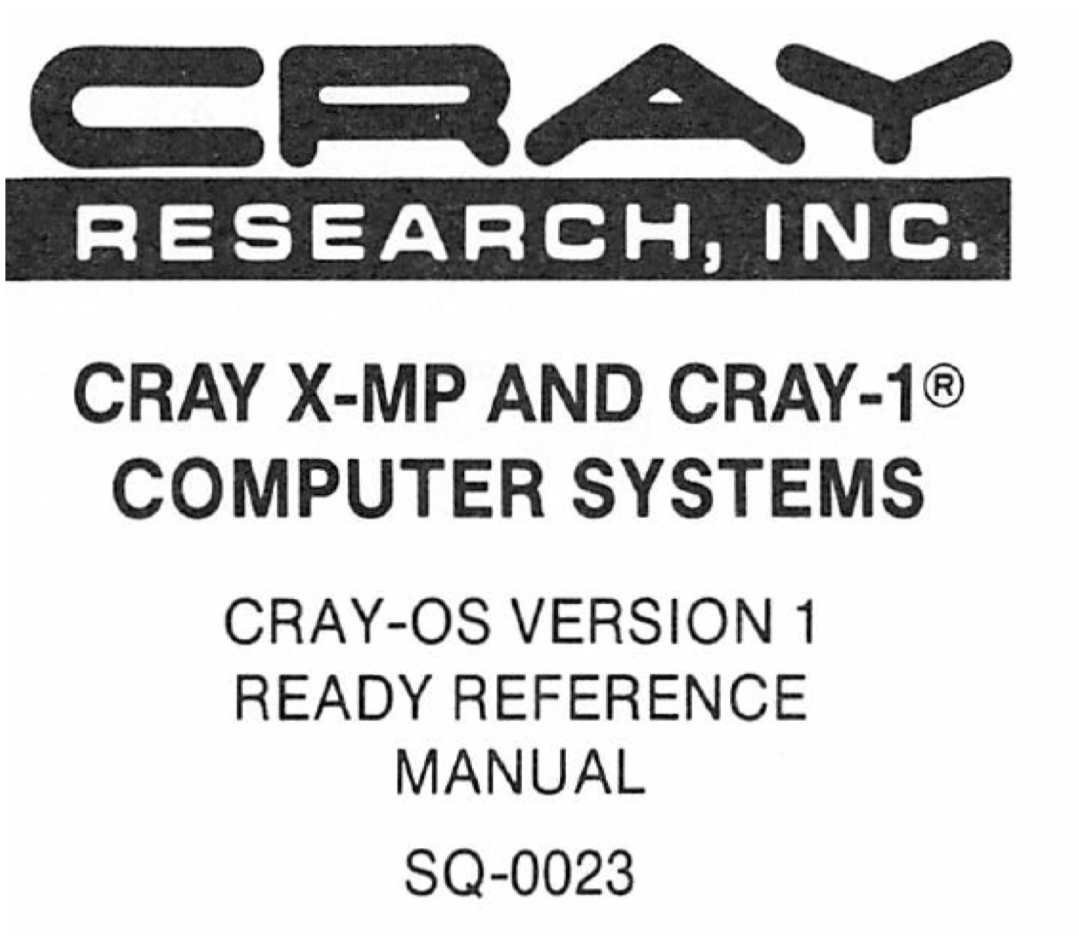
"The future is seldom the same as the past"

Seymour Cray

June 4, 1995

Seymour Cray introduced new hardware (right) in 1985 with the UNICOS operating system based on UNIX System V Release 2

Dramatic Change in Operating System

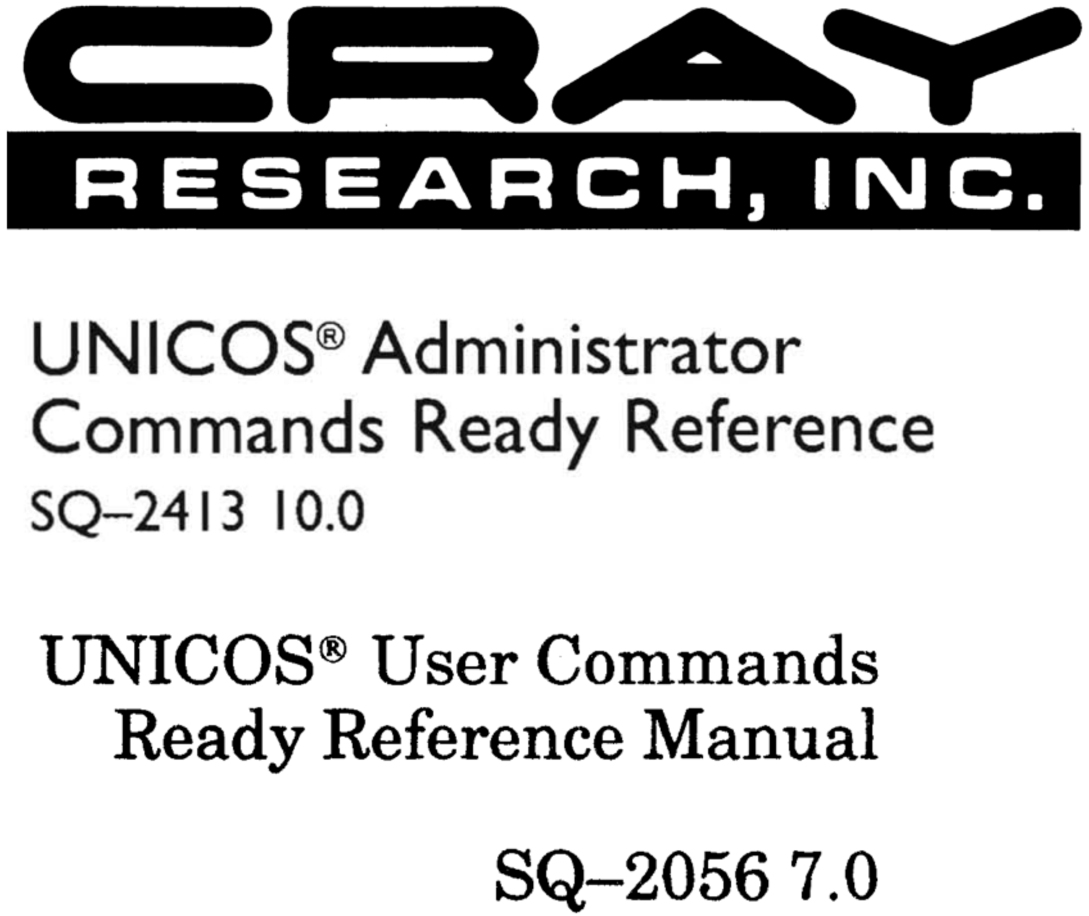


CRAY RESEARCH, INC.

CRAY X-MP AND CRAY-1®
COMPUTER SYSTEMS

CRAY-OS VERSION 1
READY REFERENCE
MANUAL

SQ-0023



CRAY RESEARCH, INC.

UNICOS® Administrator
Commands Ready Reference
SQ–2413 10.0

UNICOS® User Commands
Ready Reference Manual

SQ–2056 7.0

New Hardware, New Software

Shasta System New Hardware

- River cabinet
  - Compute and non-compute nodes in standard cabinet
- Mountain cabinet
  - High density compute nodes in larger cabinet which addresses power and cooling requirements
- Several new controllers and network switches
- New high speed network (HSN)

- Not an XC or CS system!

Shasta System New Management Software

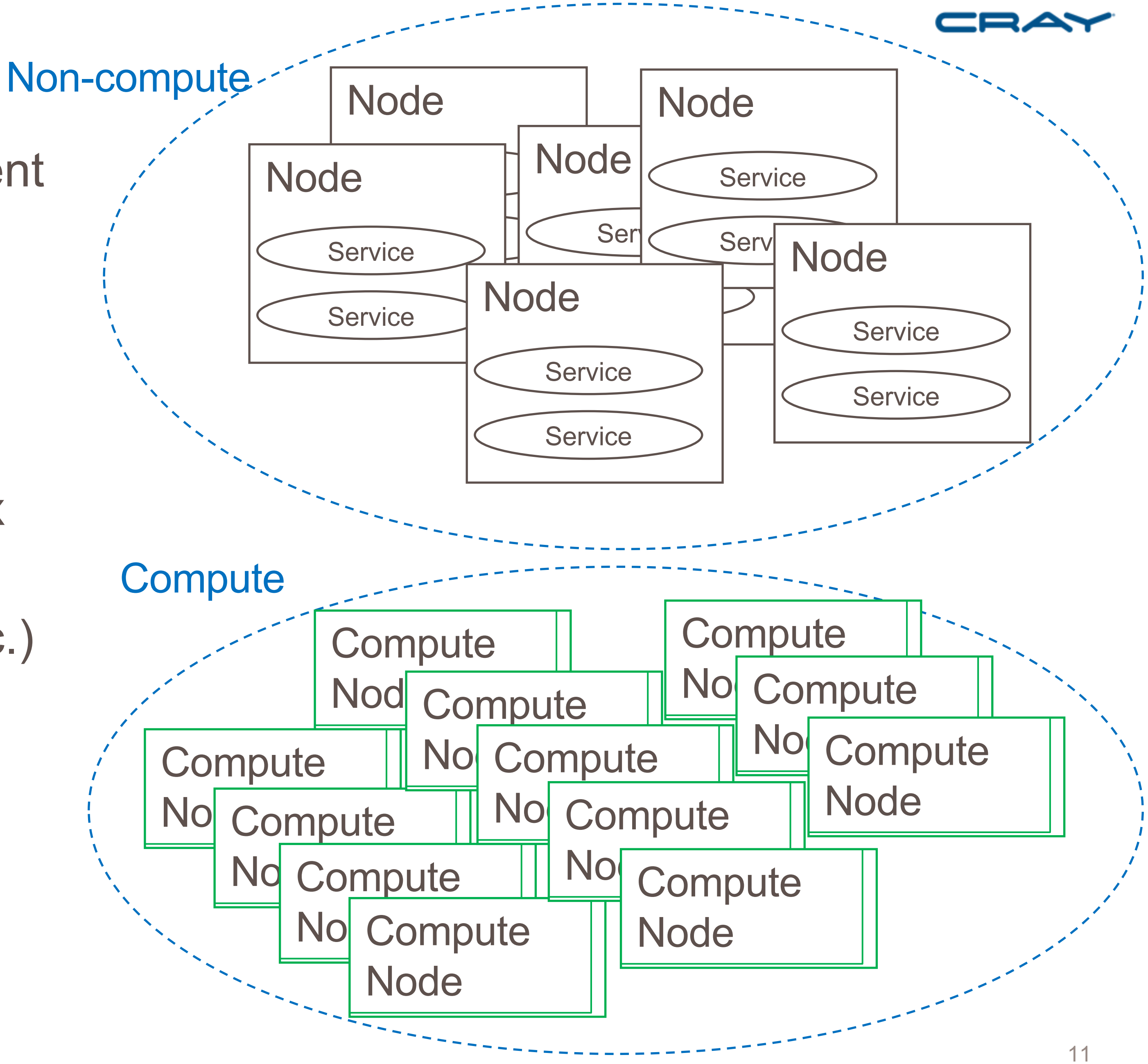
- What
  - Single Cray-authored management solution for all Shasta platform variants
- Why
  - Reduce custom software, allow standard tool use and integration with existing customer solutions
  - Support standard hardware control interfaces through Redfish
  - Ease integration of new hardware components
  - Increase flexibility to support customer-specific software orchestration and configuration strategies as required
- Benefits
  - Clean separation between management infrastructure and user-facing managed software on system
  - Resilient, Available, and Scalable by design (not an add-in “feature”)
  - Improved Security and Serviceability (hardened system management tools and environment)
  - Enhanced capabilities in telemetry management, UI, etc.

Shasta Philosophy

- How was Cray’s philosophy applied to the final solution
  - Continue to deliver performance and scale
  - Improve:
    - Availability – Multiple service instances; components upgradable without loss of service
    - Reliability – Orchestrated containers for service redundancy/resiliency
    - Usability – Common across Cray platforms with expanded functionality and UI
    - Flexibility – Use of standard and open REST APIs – extensible architecture
- What should a customer (user/sys admin) expect to see
  - Easier Cray system management, lower downtime, improved serviceability
  - Lower administrative and maintenance cost
  - Continued support for deployment and configuration of Cray’s traditional HPC software stack
  - Microservice architecture allows flexible options
    - System management services can be enabled, disabled, or extended by customer or 3rd party solutions

Shasta System Nodes

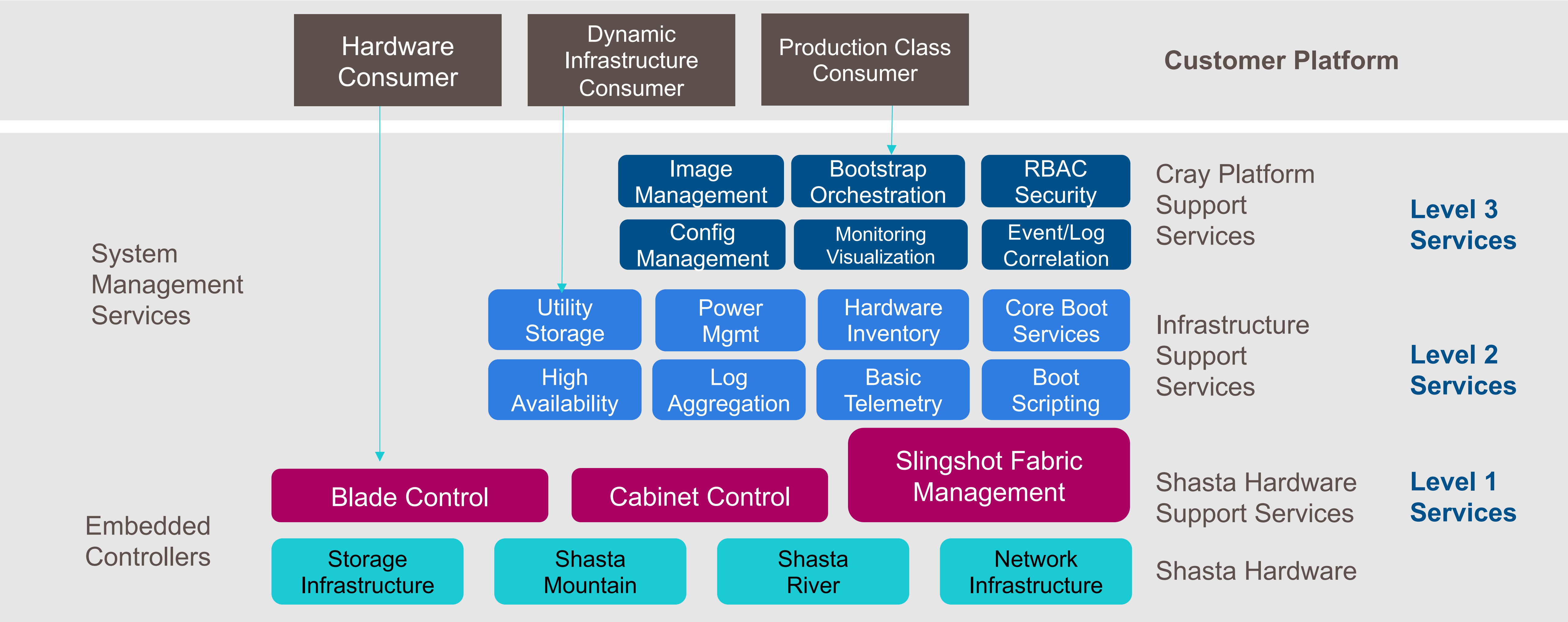
- Nodes providing services (management or managed)
  - Highly available, resilient, and scalable services
- Compute nodes
  - Shasta v1 supports full Cray Linux
  - Support for standard distributions being finalized (Centos, SLES, etc.)
  - Scheduler choice
    - WLM (Slurm, PBS, etc.)/Singularity
    - Kubernetes/Docker



Shasta System Management – Service Levels

- Service levels build on previous levels
    - Level 1 (Hardware as a Service/HaaS)
        - Provides just the software required to manage the Shasta system hardware components
            - Centered around the DMTF Redfish REST standard
            - Slingshot Fabric and Network management
    - Level 2 (Infrastructure as a Service/IaaS)
        - Provides infrastructure support for deploying basic software stacks onto the system
        - Centered around bootstrapping a customer supplied image
    - Level 3 (Platform as a Service/PaaS)
        - Provides full support for deployment of Cray custom capability software stack at scale
        - Familiar features and functions to Cray XC but
            - Enhanced scalability, resiliency, and security, for both Shasta Software Stack and customer host environments

Shasta System Management



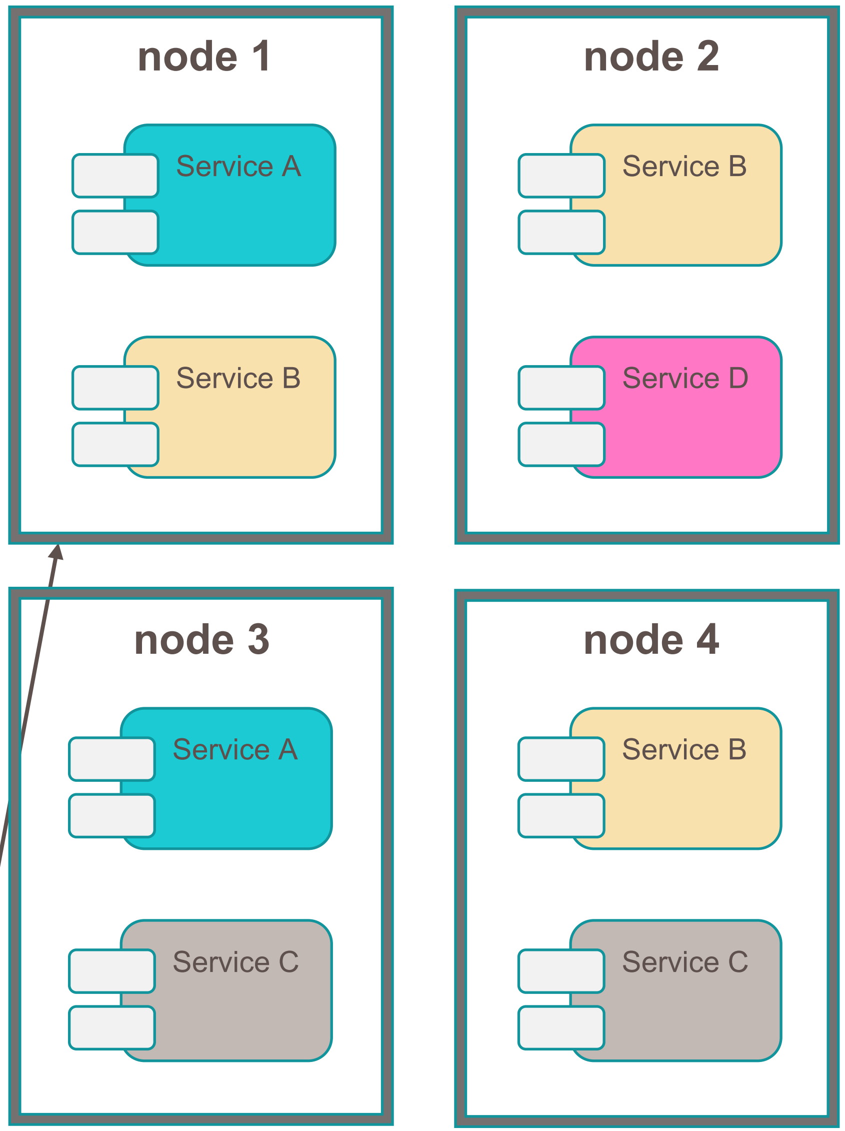
Microservices with REST APIs

Service Based Architecture

- Services
  - Represent a logical activity within the system
  - Are self-contained
  - Only expose interfaces (or APIs) for communication with other services and components
- Modular approach
  - Decouples the services from each other
  - Allows for greater ease of maintenance and replacement of the components within each service
  - As long as the versioned API behaves the same, there is no need for another service or component that relies on it to know its internal structure or implementation

Distribute Services

- Compose an application by integrating distributed, separately-maintained, and deployed software components
- Enabled by technologies and standards that make it easier for components to communicate and cooperate over a network
- Increases the reliability, availability, and scalability of the management functions
- Enables scaling across multiple hosts
- Allows the system management requests to be load balanced across a distributed system for automatic scalability and reliability



Multiple non-compute nodes distribute service load

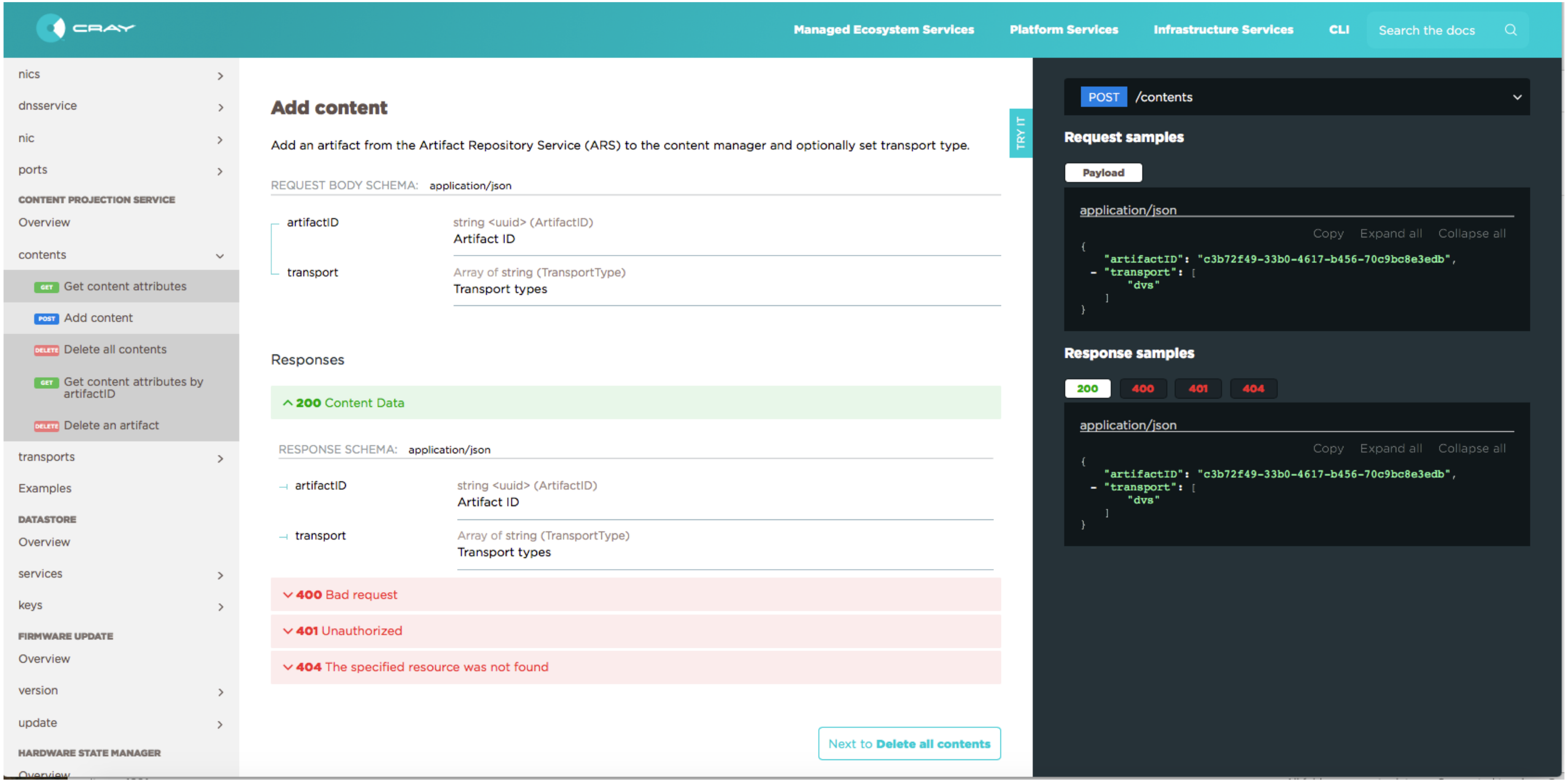
Data Model for Microservices

- Each service owns its own data store
  - Uses persistent storage provided by Utility Storage
- Access from other services is via a REST API
  - Pub/sub model can be used
- Advantages:
  - Can scale without impacting other services' state
  - Can change the underlying data store without affecting other services
  - Can update service without dependency on other services

REST API

- A RESTful API is an application program interface (API) that uses HTTP requests
  - GET, DELETE, PUT, PATCH, POST
- REST API specification (swagger/OpenAPI 3.0) for Cray microservices used to generate
  - API documentation
    - Provided in docker image and in tarball for webserver
  - API server stubs for the microservice
  - API client code for the Cray CLI framework

API Documentation from REST API Specification



Add content

Add an artifact from the Artifact Repository Service (ARS) to the content manager and optionally set transport type.

REQUEST BODY SCHEMA: application/json

artifactID	string <uuid> (ArtifactID) Artifact ID
transport	Array of string (TransportType) Transport types

Responses

200 Content Data

RESPONSE SCHEMA: application/json

artifactID	string <uuid> (ArtifactID) Artifact ID
transport	Array of string (TransportType) Transport types

400 Bad request

401 Unauthorized

404 The specified resource was not found

Next to **Delete all contents**

POST /contents

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "artifactID": "c3b72f49-33b0-4617-b456-70c9bc8e3edb",
  - "transport": [
        "dvs"
    ]
}
```

Response samples

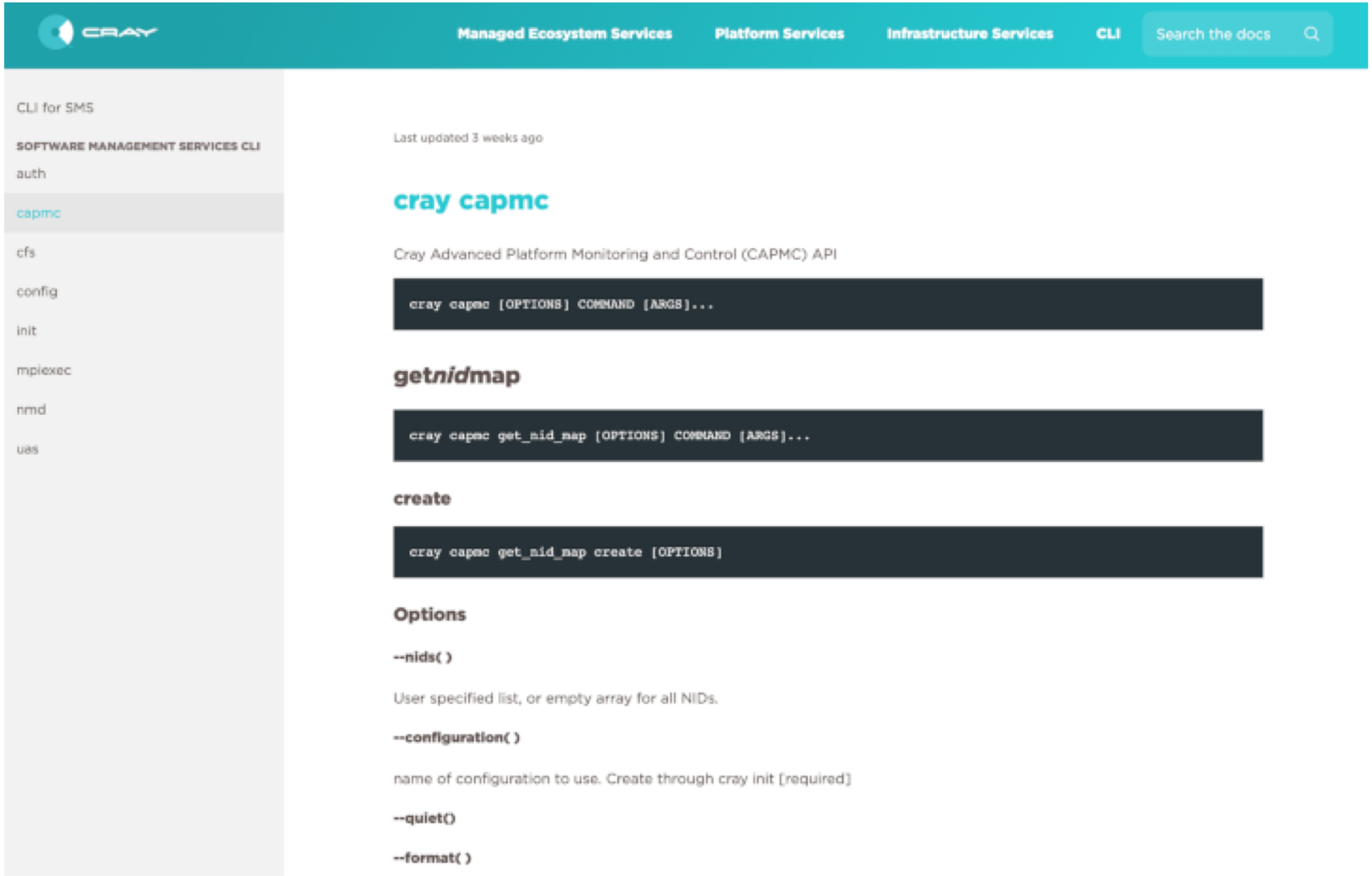
200 400 401 404

application/json

Copy Expand all Collapse all

```
{
    "artifactID": "c3b72f49-33b0-4617-b456-70c9bc8e3edb",
  - "transport": [
        "dvs"
    ]
}
```

CLI Documentation from REST API Specification



Last updated 3 weeks ago

cray capmc

Cray Advanced Platform Monitoring and Control (CAPMC) API

```
cray capmc [OPTIONS] COMMAND [ARGS]...
```

get*nid*map

```
cray capmc get_nid_map [OPTIONS] COMMAND [ARGS]...
```

create

```
cray capmc get_nid_map create [OPTIONS]
```

Options

--nids( )

User specified list, or empty array for all NIDs.

--configuration( )

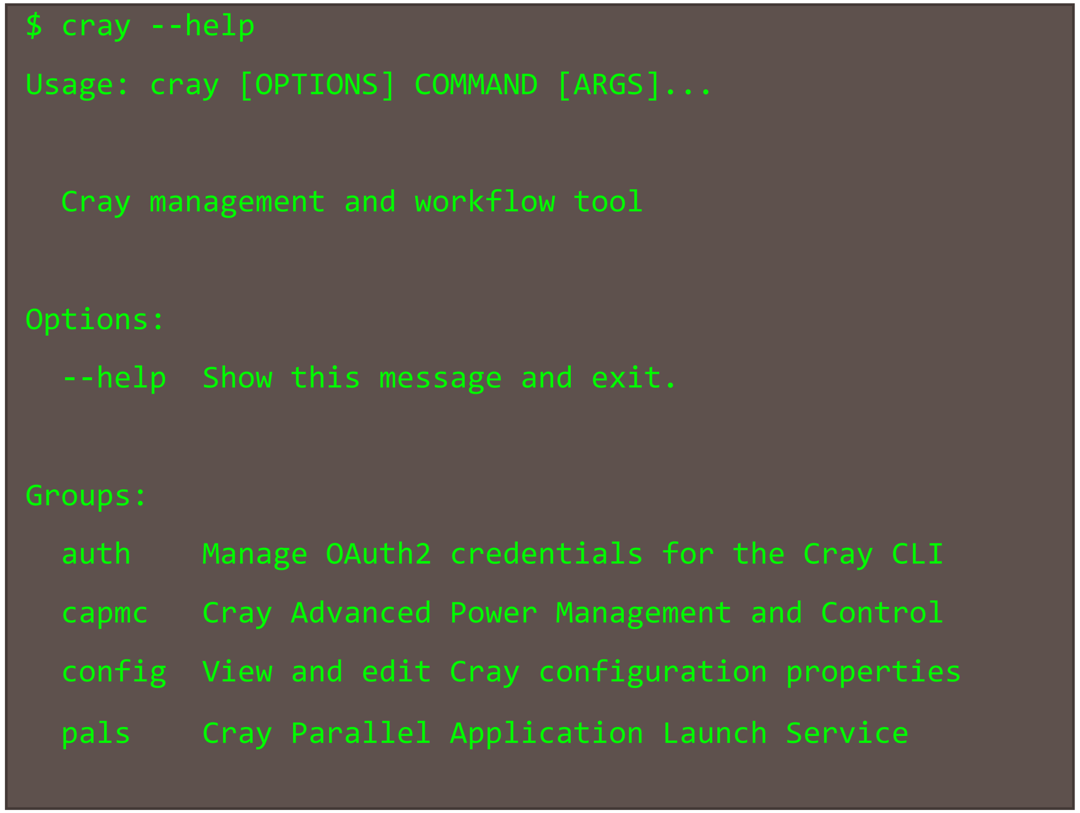
name of configuration to use. Create through cray init [required]

--quiet()

--format( )

CLI Framework from REST API Specification

- New CLI for interacting with Shasta Management
  - Based on REST APIs and minimal code
  - Generated CLI
  - Built on a set of open standards
  - REST for all control



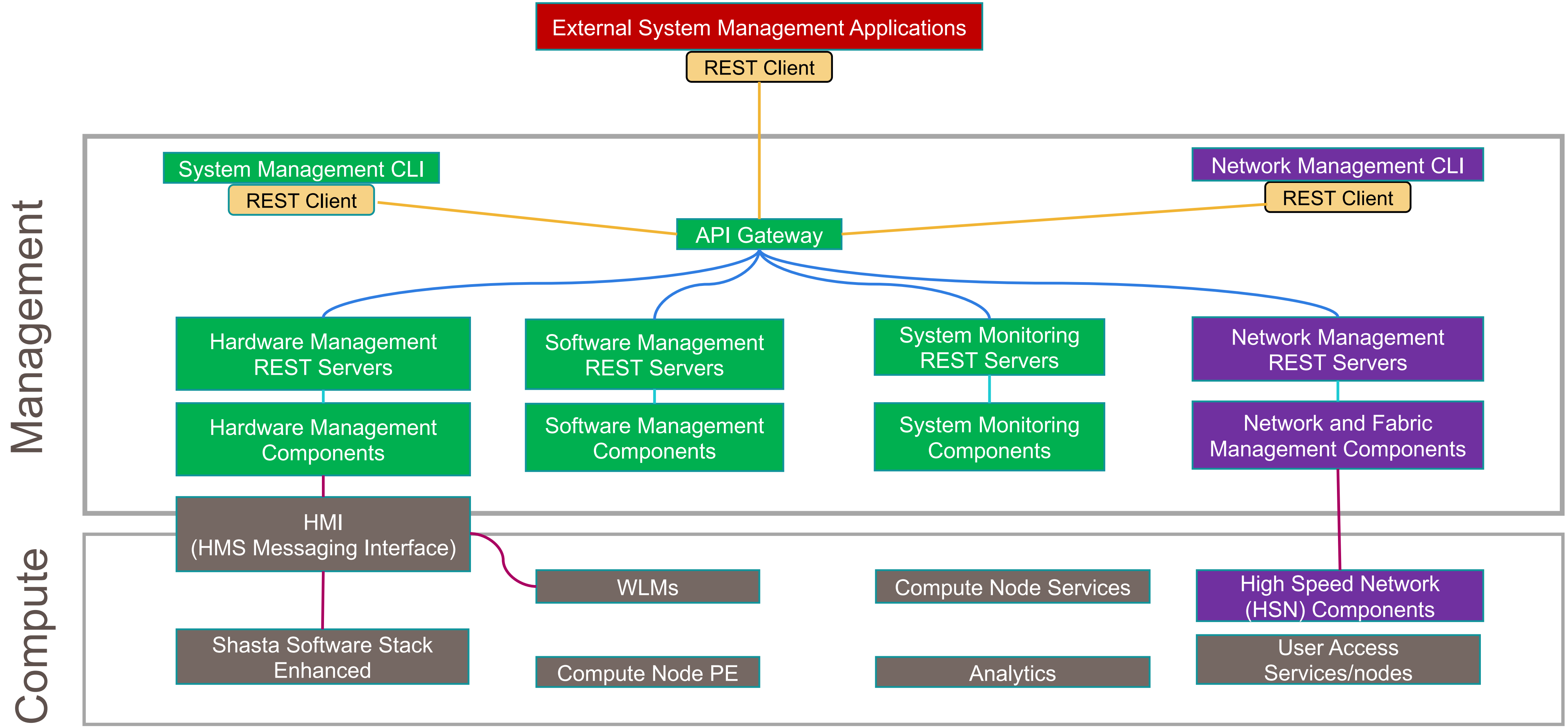
```
$ cray --help
Usage: cray [OPTIONS] COMMAND [ARGS]...

  Cray management and workflow tool

Options:
  --help  Show this message and exit.

Groups:
  auth    Manage OAuth2 credentials for the Cray CLI
  capmc   Cray Advanced Power Management and Control
  config  View and edit Cray configuration properties
  pals    Cray Parallel Application Launch Service
```

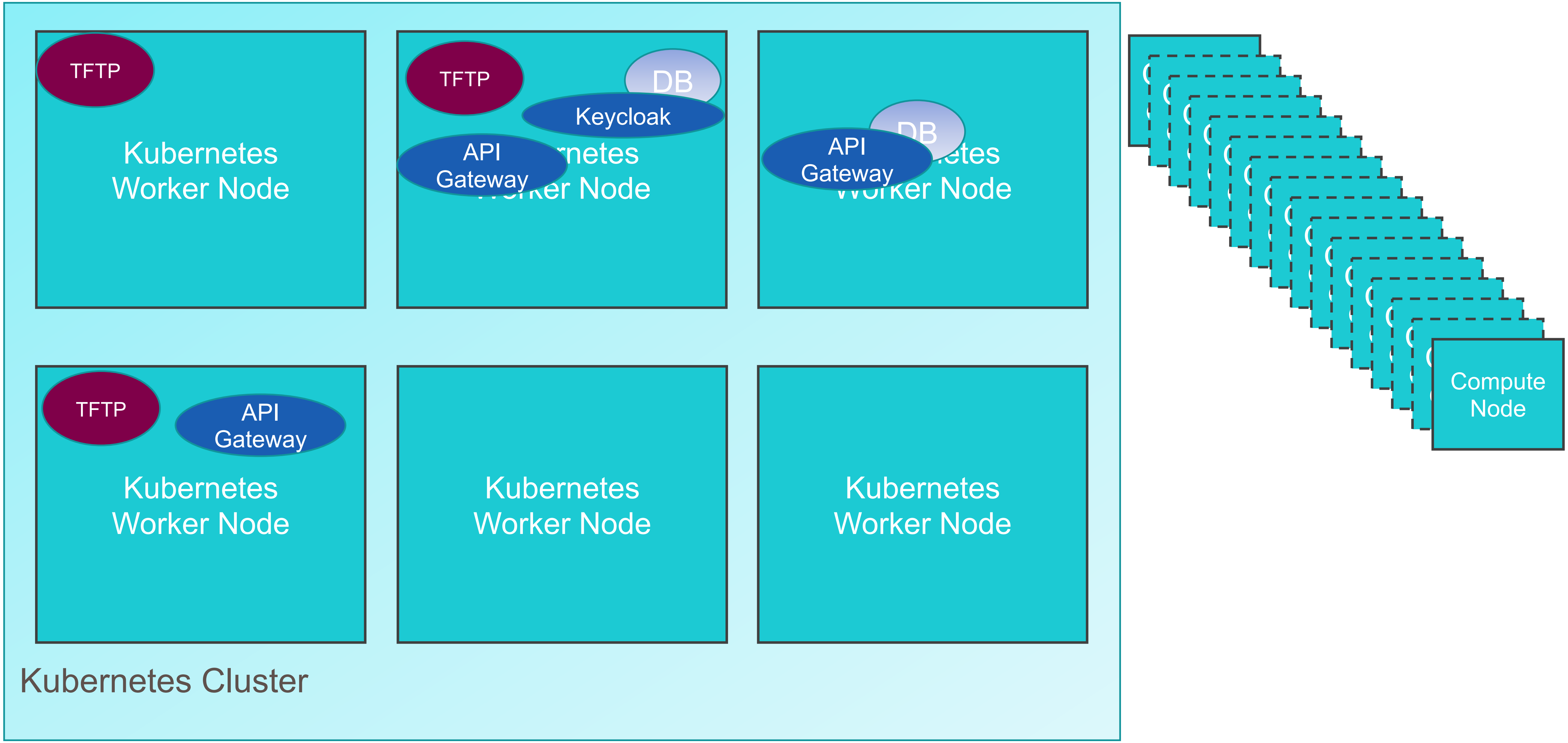
System Management API Gateway


External System Management Applications
REST Client
Management
System Management CLI
REST Client
Network Management CLI
REST Client
API Gateway
Hardware Management REST Servers
Software Management REST Servers
System Monitoring REST Servers
Network Management REST Servers
Hardware Management Components
Software Management Components
System Monitoring Components
Network and Fabric Management Components
HMI (HMS Messaging Interface)
Compute
WLMs
Compute Node Services
High Speed Network (HSN) Components
Shasta Software Stack Enhanced
Compute Node PE
Analytics
User Access Services/nodes

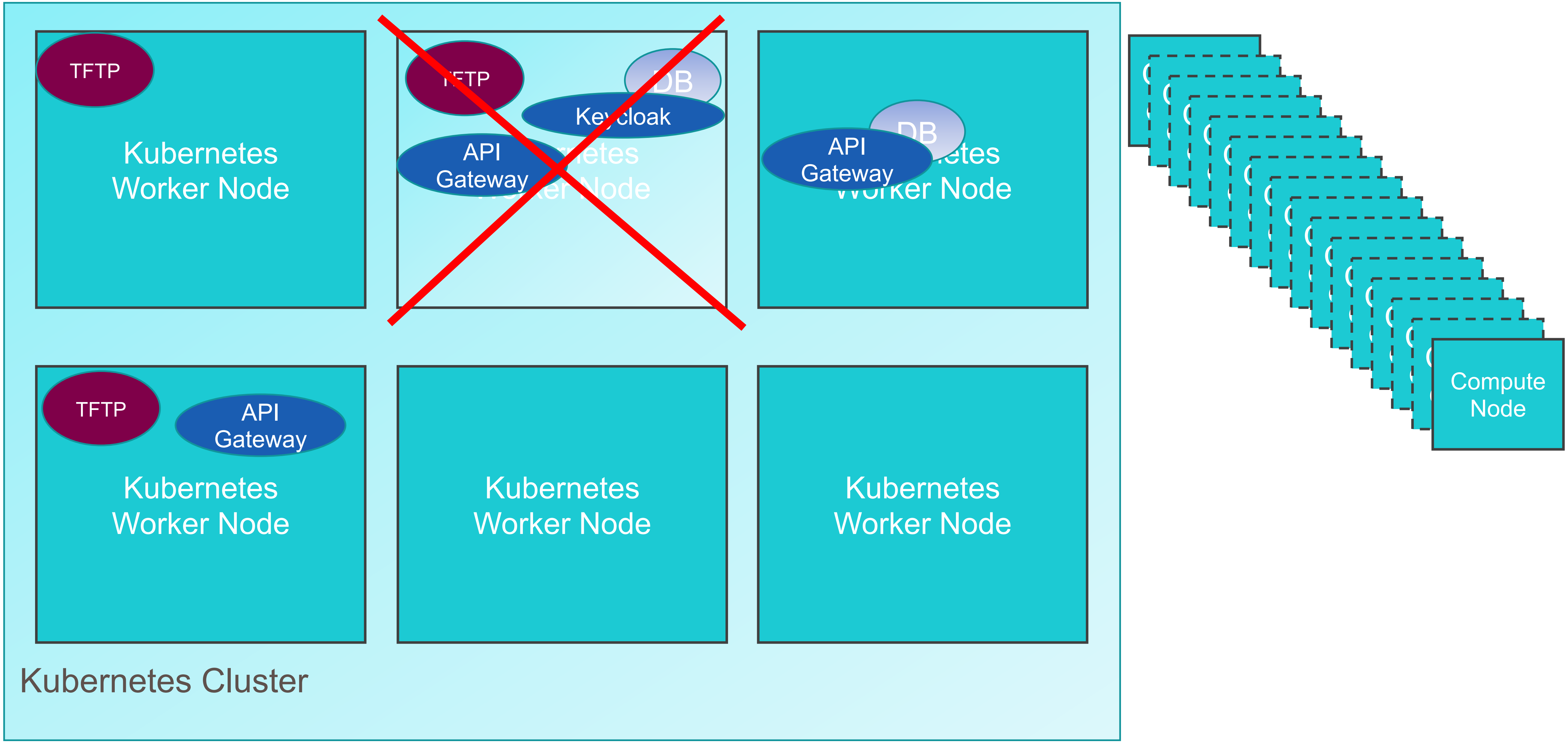
Docker and Kubernetes

- Docker
  - Docker container runtime
  - Docker execution environment
    - Standardizes the management
  - Configuration data passed into the container modules
    - Code that provides the networking is the same for every container
- Kubernetes
  - Manages the life cycle of containers within the service infrastructure
    - Scheduling of containers to run across a set of hosts
    - Controlling where to run service based on requirements of the service
    - DNS and networking support between containers in a system
    - Automatic scaling and health monitoring
    - Upgrade strategies

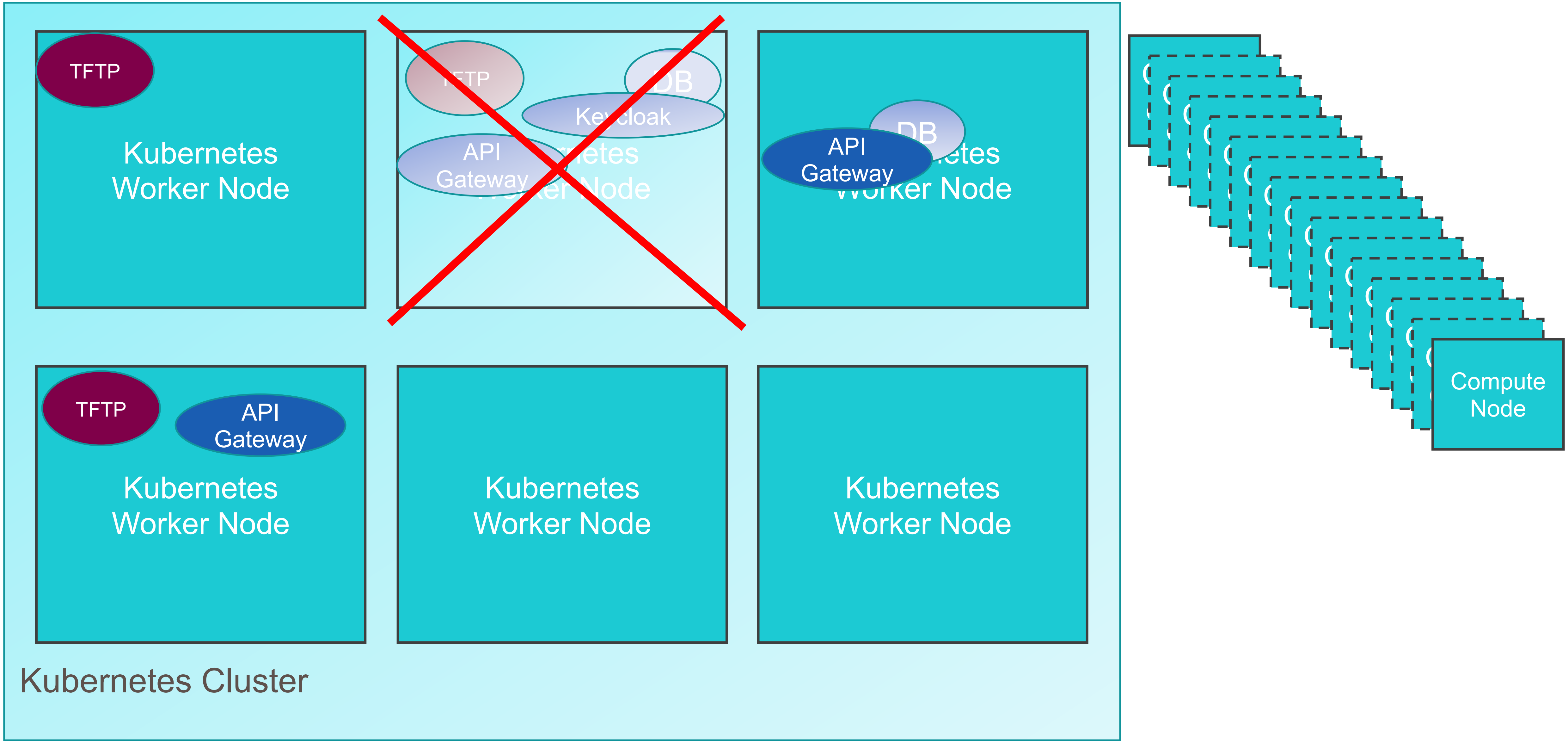
Resiliency with Kubernetes – Services Running



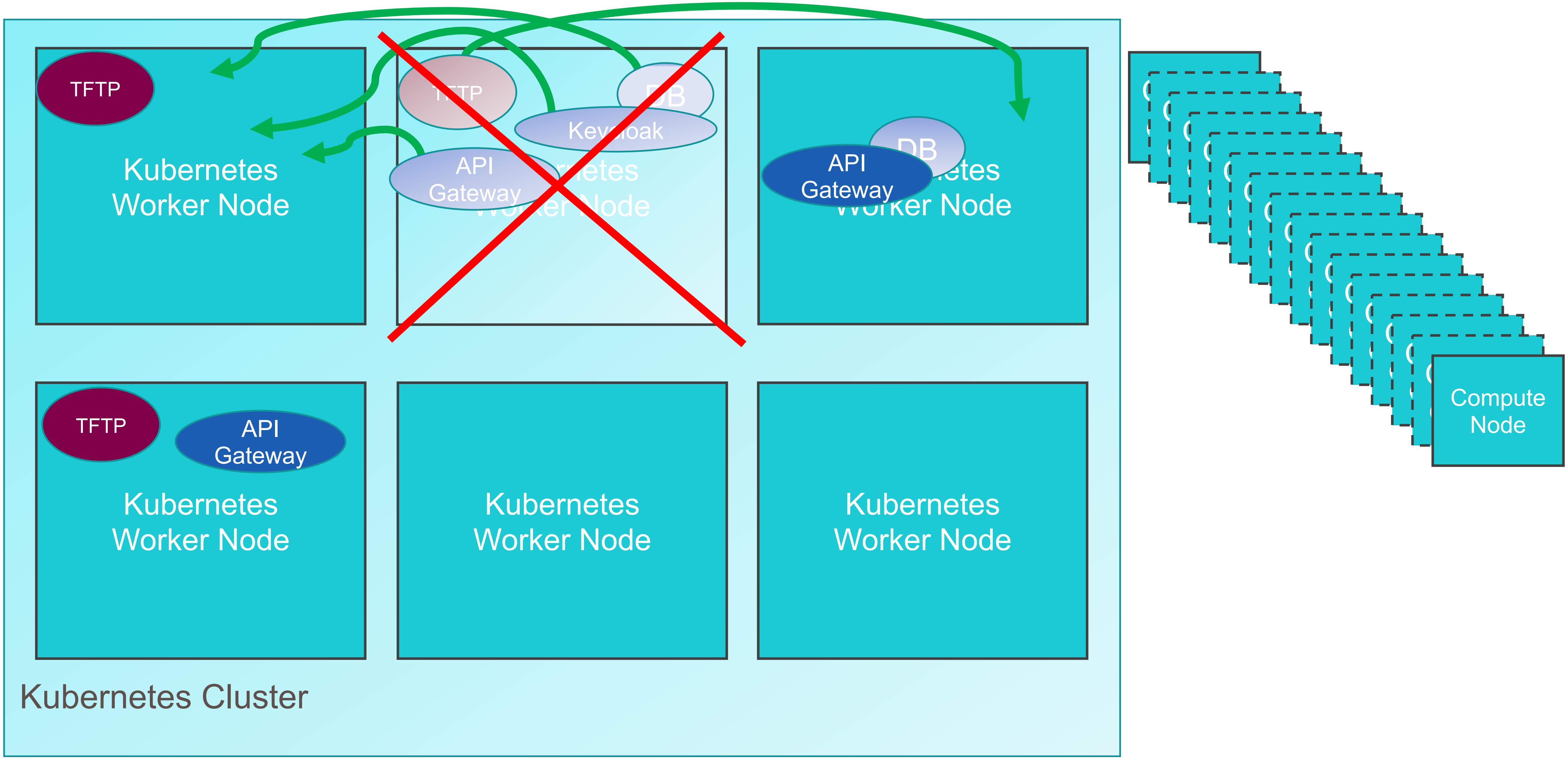
Node Goes Down



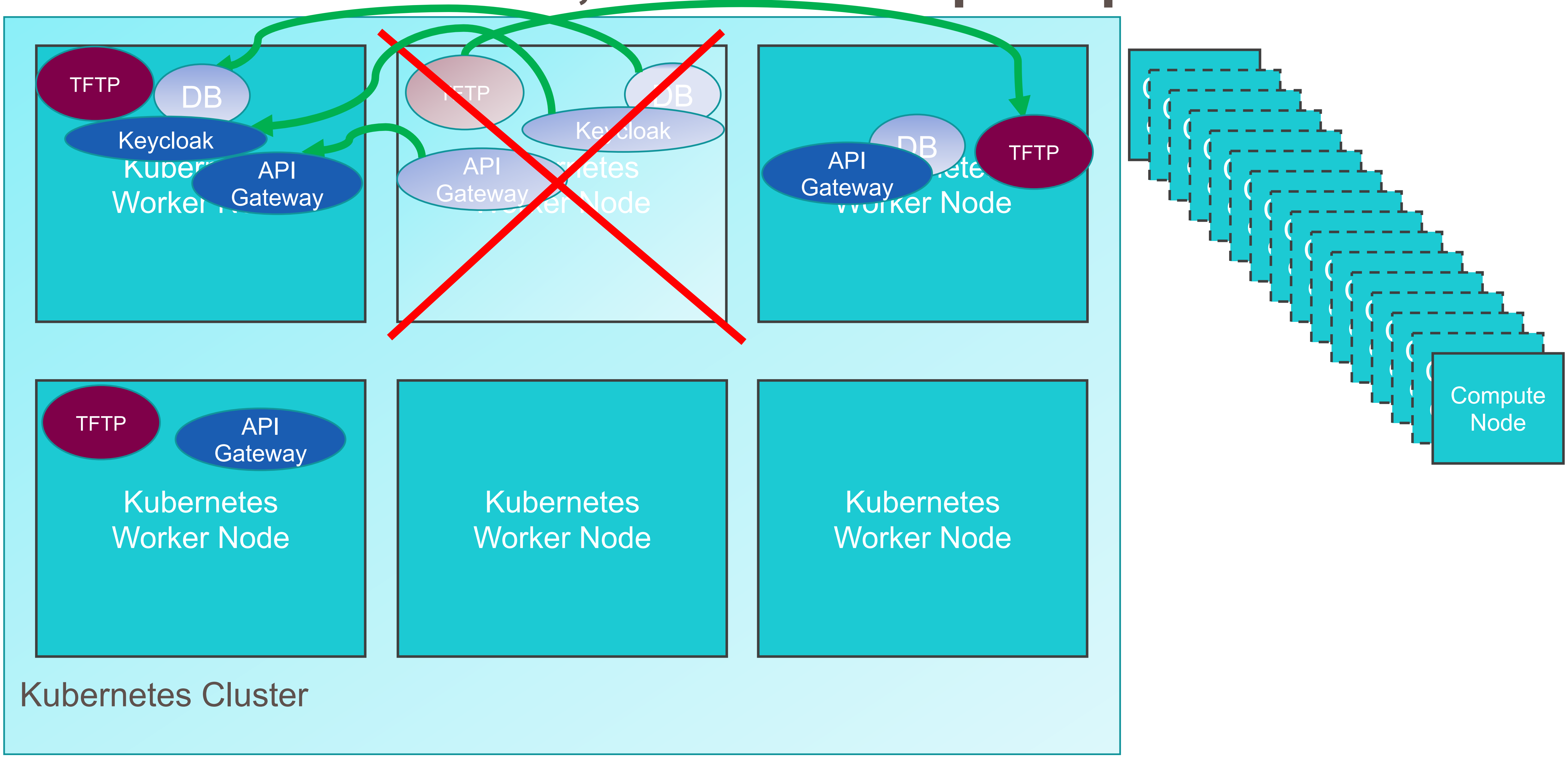
Node and Services Stop Responding



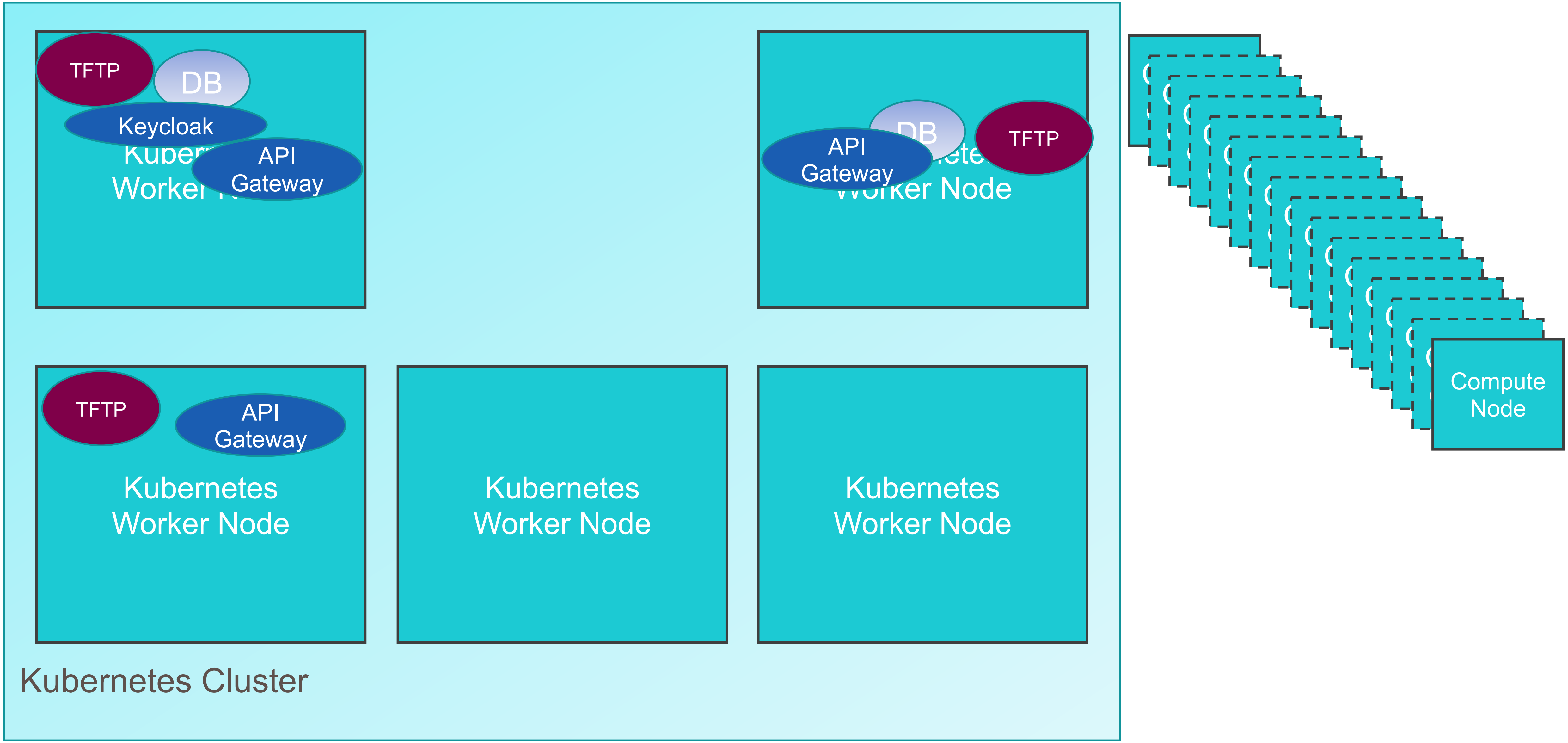
Kubernetes Selects New Nodes for Pods



Old Pods Terminate, New Pods Spin Up



Node Removed From Kubernetes Cluster



© 2019 Cray Inc.

Management Functionality

Shasta System Management Comparison

Hardware IaaS Level 2

- 🟦 System Discovery
- 🟦 HW State Mgmt.
- 🟦 HW Inventory Mgmt.
- 🟦 Power Control
- 🟦 Environment Control
- 🟦 Firmware Mgmt.
- 🟦 Event Notification Services
- ▨ Node BIOS

Software IaaS Level 2

- 🟪 SW Inventory Mgmt.
- 🟦 Scalable Boot Services
- 🟦 Image Mgmt.
- 🟦 Node/Image Mapping
- 🟪 HSN Fabric Mgmt.
- 🟪 Infrastructure Telemetry System
- 🟪 Utility Storage
- 🟪 Network Mgmt.
- 🟪 Managed SW State Tracking

PaaS Level 3

- 🟦 Boot orchestration Services
- ▨ SW Configuration Mgmt. (Ansible)
- 🟦 Image Creation
- 🟪 Content Repository
- 🟪 Advanced HSN Fabric Controls
- 🟦 Workload Manager Interfaces

Supporting Services

- 🟦 High Availability
- 🟦 Data Services
- 🟪 Telemetry Bus
- 🟪 GUI Monitoring & Analysis
- ▨ Node Debug
- 🟦 Fabric Debug
- 🟦 Online Diagnostics
- ▨🟦 GPU Management

Extensibility Interoperability

- 🟪 Hardware Services API
- 🟪 Redfish
- 🟪 Infrastructure Services API
- 🟪 Platform Services API
- 🟪 External Telemetry Bus

🟦 CLE on XC ▨ 3rd party with possible Cray added value 🟪 New for Shasta systems

Shasta Presentations at CUG

Shasta Presentations at CUG 2019

- Hardware
  - Shasta Hardware Workshop
- Software
  - Shasta Software Workshop
  - Shasta System Management Overview (this presentation)
  - Shasta System Monitoring Framework
  - Reimagining Image Management in the New Shasta Environment
  - Hardware Discovery and Maintenance Workflows in Shasta Systems
  - Resource Management in a Heterogeneous Environment
  - The role of emerging orchestration and execution models in HPC Environments
- Customer experience with early Shasta systems
  - Exploring New Monitoring and Analysis Capabilities on Cray's Software Preview System
  - Exploring the Mysterious Universe of Shasta Software for Perlmutter
- Shasta Presentations at CUG 2018
  - Cray Next Generation Software Integration Options
  - Modernizing Cray Systems Management – Use of Redfish APIs on Next Generation Shasta Hardware

SAFE HARBOR STATEMENT

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts.

These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

QUESTIONS?

✉ Harold Longley

cray.com

@cray_inc

linkedin.com/company/cray-inc-/