IBM and Red Hat Forum

IBM Crypto Express meets RHOCP on IBM Z/LinuxONE

Reinhard Bündgen
buendgen@de.ibm.com
Hendrik Brückner
brueckner@de.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*	CICS*	IBM Cloud*	IMS	z/VM*
ibm.com	Db2*	IBM Cloud Pak*	Power*	
IBM (logo)*	HiperSockets	IBM Z*	z/OS*	

*** Registered trademarks of IBM Corporation**

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.
ITIL is a Registered Trade Mark of AXELOS Limited.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.
RStudio®, the RStudio logo and Shiny® are registered trademarks of RStudio, Inc.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.
Zowe™, the Zowe™ logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.
Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Abstract

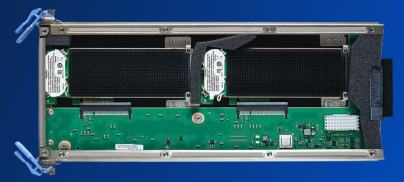
Kubernetes (K8s) and Red Hat OpenShift Container Platform (RHOCP) misses the capabilities to allow containerized application (in POD) to use cryptographic functions provided by IBM Crypto Express cards on IBM Z & LinuxONE.
The presentation describes a new extension to Kubernetes and RHOCP that allows a container running in a POD to access a domain of a Crypto Express adapter. The presentation describes how to set up the cluster infrastructure, how to configure which crypto resource belongs to which workload, and how software inside the container can access the crypto resource. The enablement requires a new component on the compute nodes which is called the Kubernetes device plug-in for IBM Crypto Express cards which is now available as a certified container image in the Red Hat catalogue.

Introduction / Motivation

IBM Z and LinuxONE platform is designed for security.

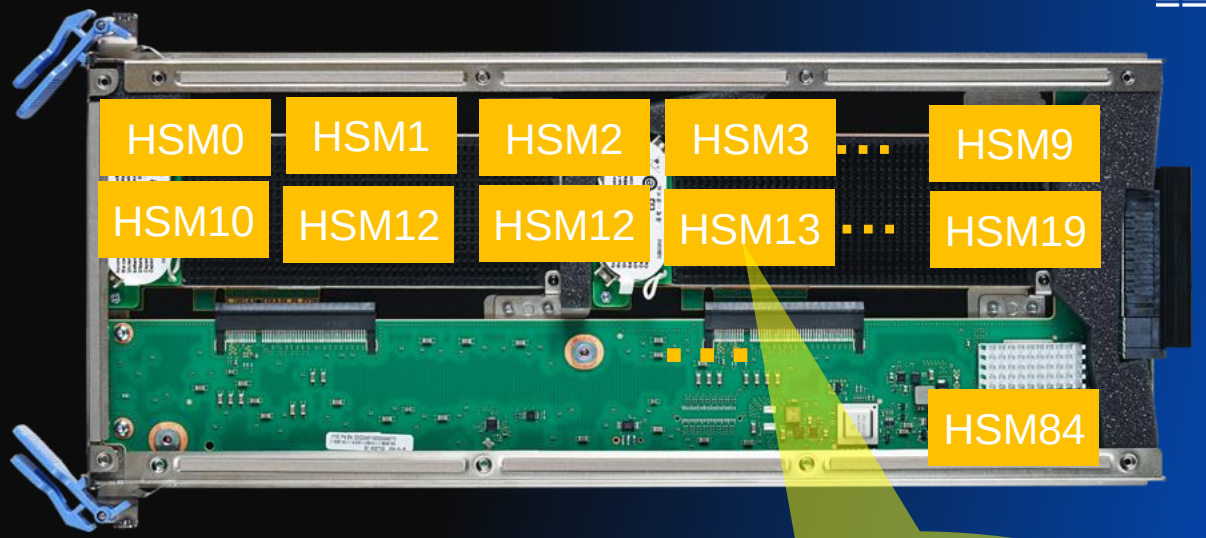
IBM Crypto Express adapters provide

- RSA acceleration designed for high TLS throughput
- FIPS 140-2 level 4 certification
- virtualizable CCA and EP11 HSMs



Goal: allow RHOCP (K8s) containers to use IBM Z crypto resources

Crypto Express Adapters



Three different firmware loads

- Accelerator mode
- CCA mode -> HSM
- EP11 mode (since CEX4S) -> HSM

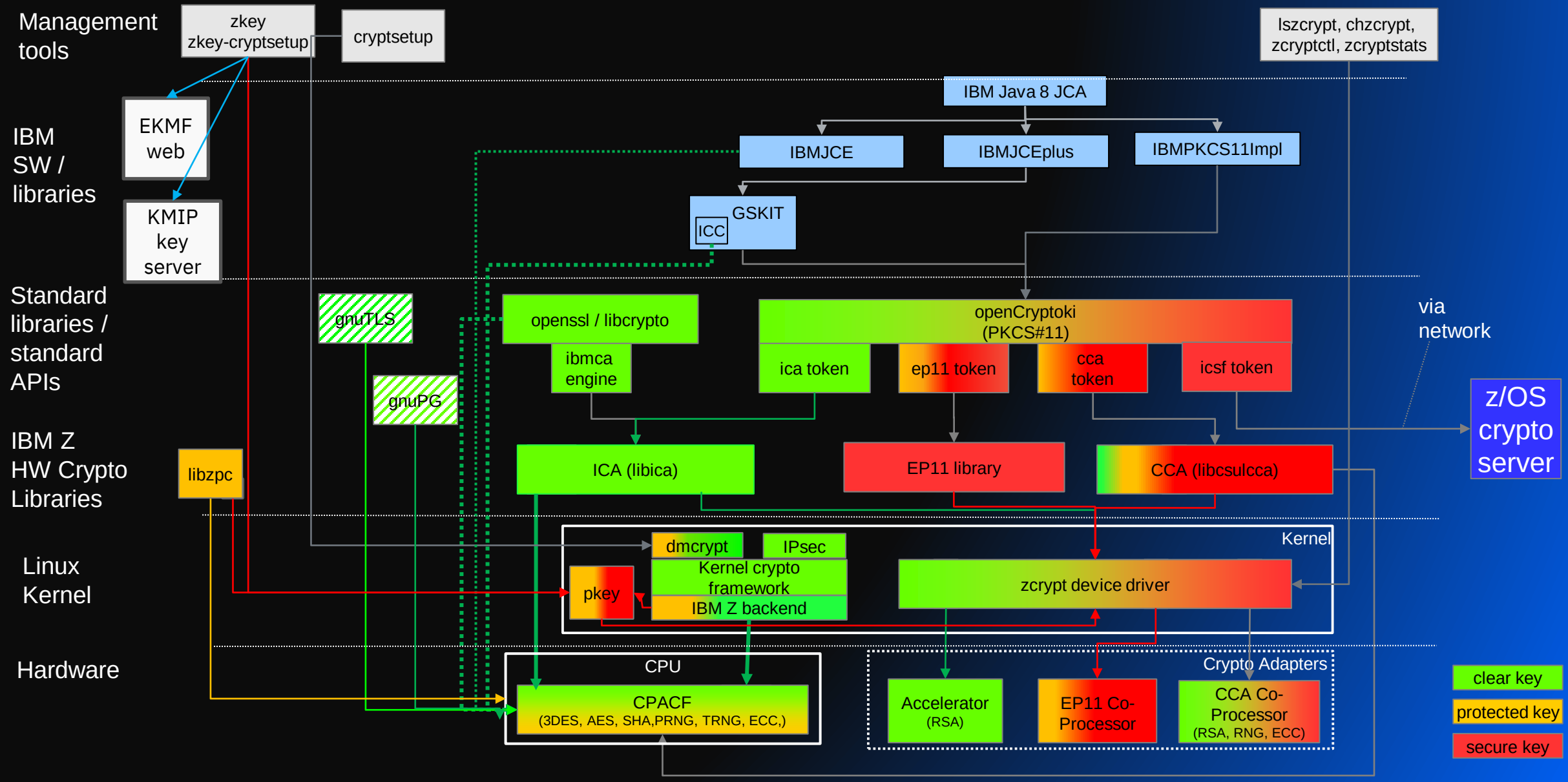
Adapter virtualization

- Adapter can be partitioned into domains of the same mode (separate master keys per domain)
- CEX5S/6S/7S:
  - 85 domains on z13, z14, z15, LinuxONE (II,III) Emperor/LT1
  - 40 domains on z14 ZR1, z13s, z15 T2, LinuxONE (II, III) Rockhopper/LT2
- within a system, each adapter domain is addressed by an APQN, a pair of an adapter ID and a domain ID

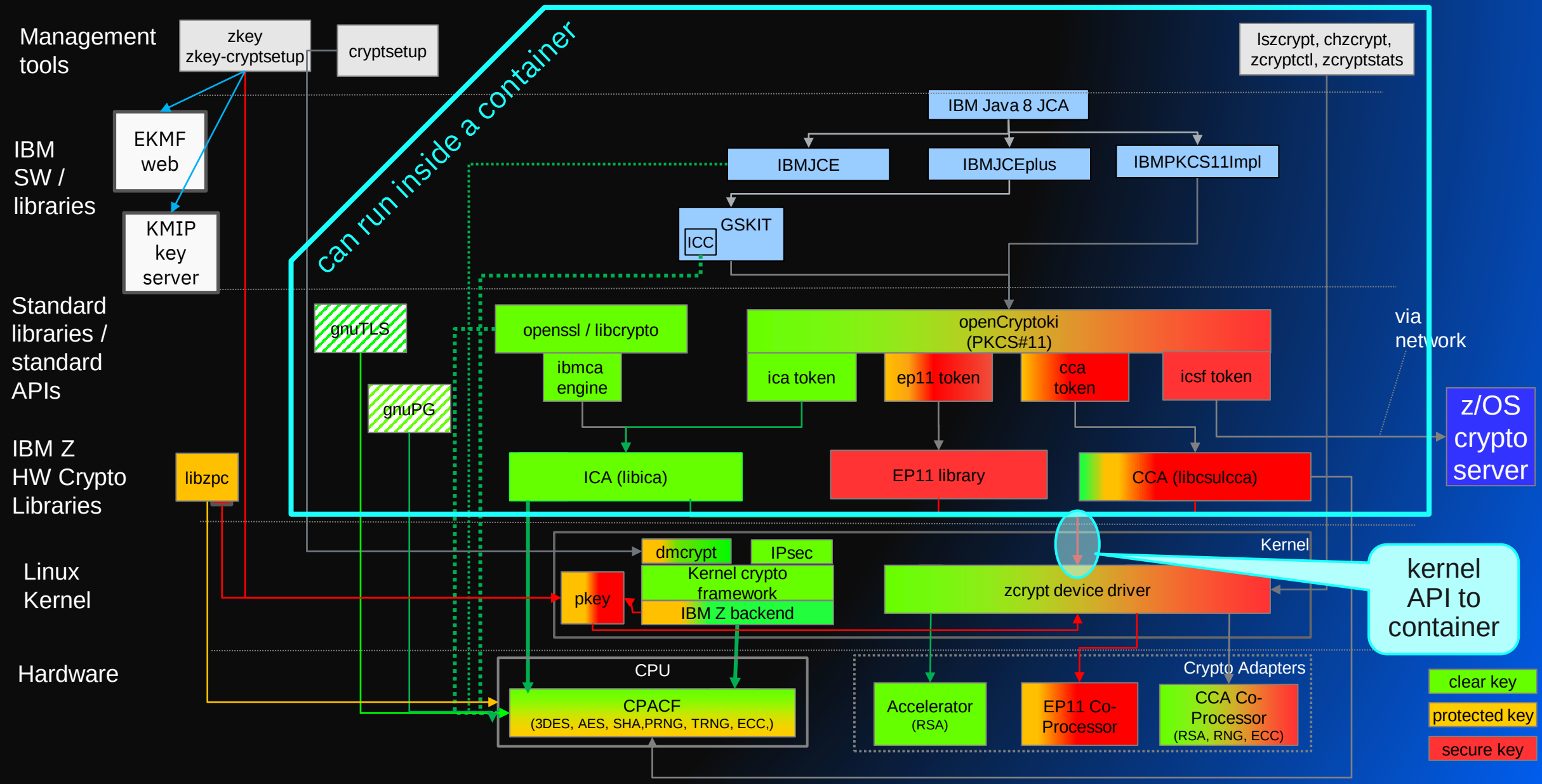
Adapter management via Support Element (SE) or HMC

- SE: Selection of adapter mode (firmware load)
- HMC: Assignment of adapters and domains to LPARs
  - usage domains: targets for crypto operation
  - control domains: targets for management operations (e.g., master key management)

The Linux on Z and LinuxONE crypto stack



The Linux on Z and LinuxONE crypto stack



Red Hat OpenShift Container Platform (RHOCP)

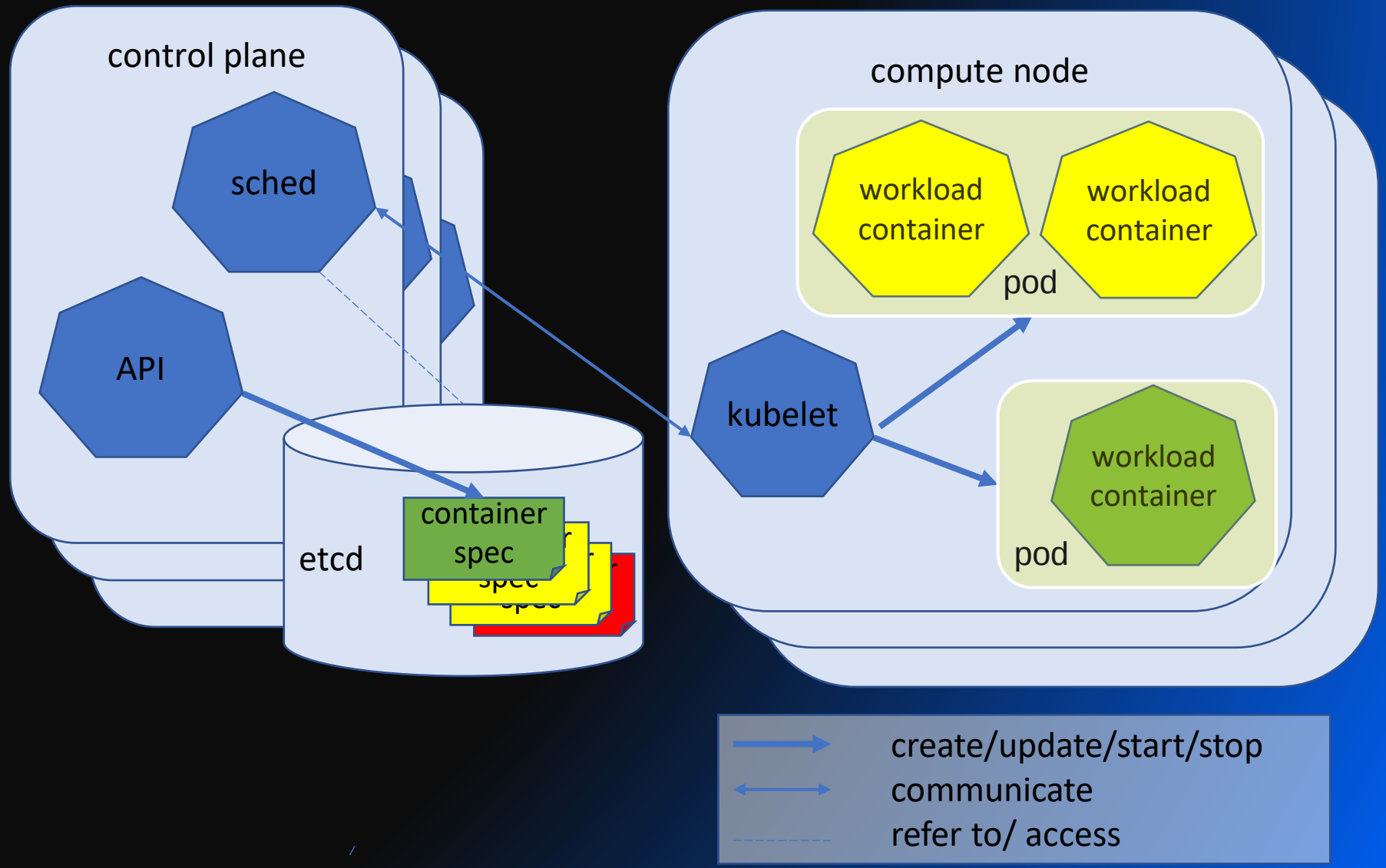
http://public.dhe.ibm.com/software/dw/linux390/docu/RHOCP-reference-architecture.pdf

A cluster framework to deploy workloads as containers

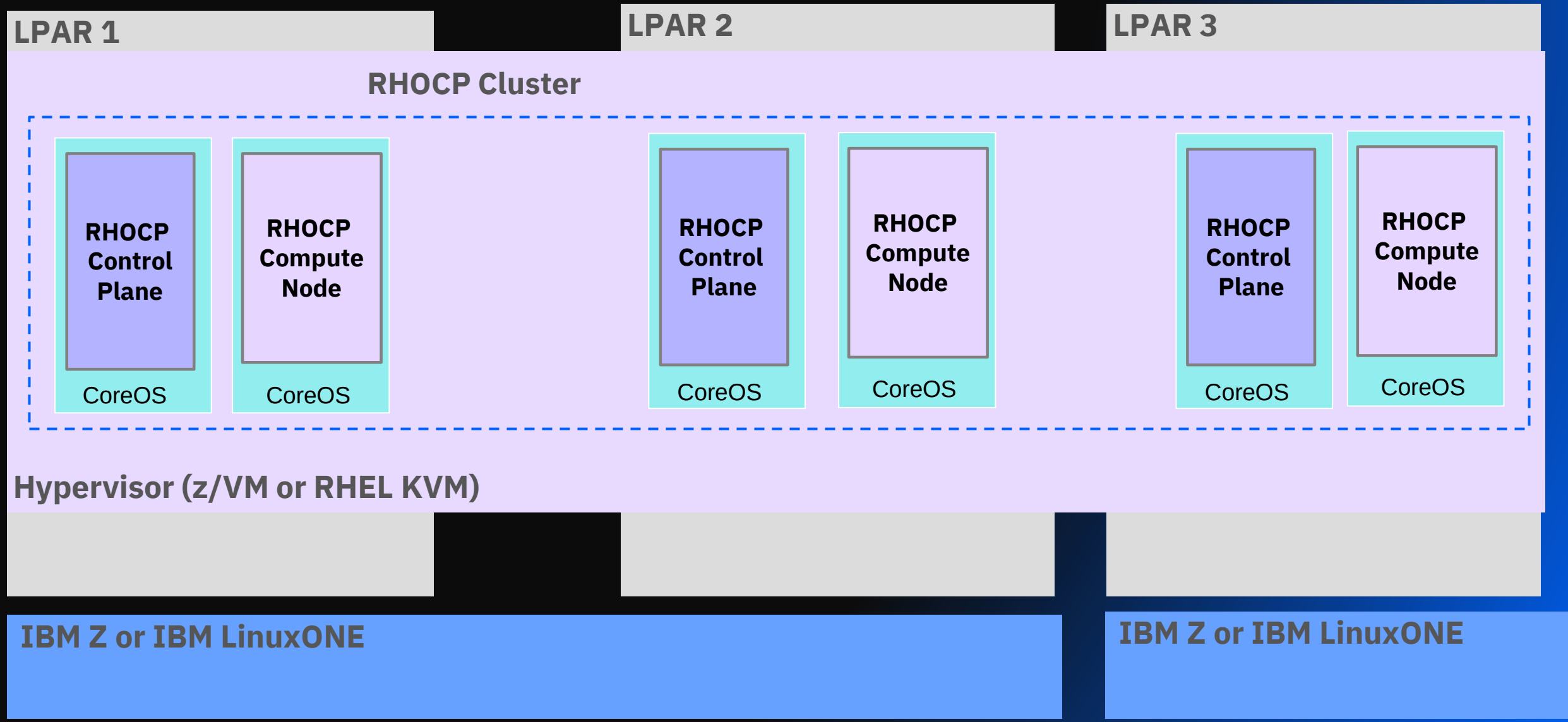
- high availability
- flexible deployment
- based on Kubernetes (K8s) technology

nodes

- control planes aka management nodes
- compute nodes run actual workload
- node OS: RHEL CoreOS
- pod: set of containers implementing a workload



Sample production environment across two Z/LinuxONE servers



IBM Z and LinuxONE Crypto HW Support

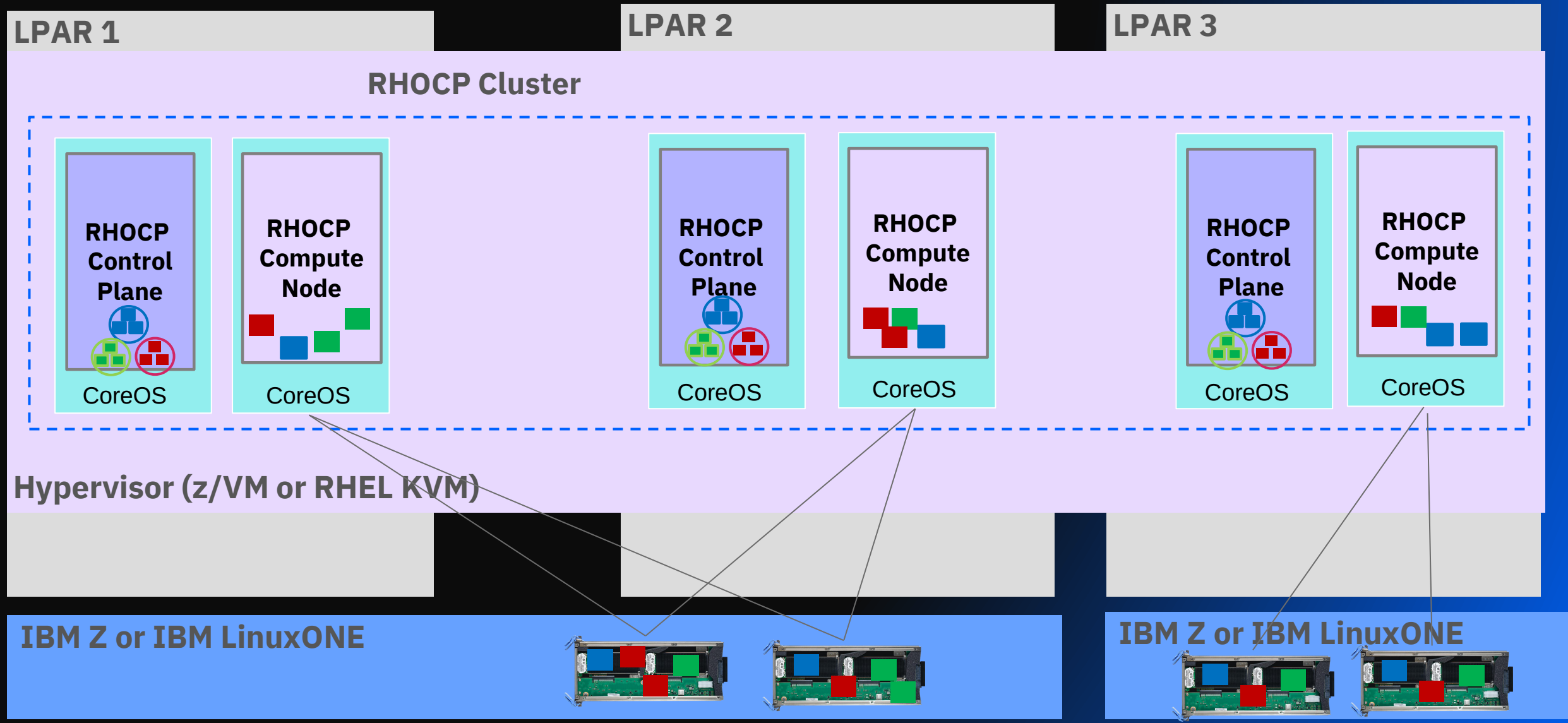
Usage of CPACF (by kernel crypto, openSSL, IBM Java 8 JCE, Go, ...) transparent in

- control plane
- compute node components
- containers running in compute nodes

Usage of Crypto Express adapters

- not supported by Kubernetes
- requires platform specific enablement
- here: Release 1.0 of this enablement

Sample production environment across two Z/LinuxONE servers



Configuring CEX adapters for Kubernetes clusters

Crypto Express cards must be plugged into IBM Z or LinuxONE hardware hosting compute nodes

– The mode of each crypto module (adapter) must be configured on the SE

An LPAR running a hypervisor (KVM or z/VM)

– must be assigned a list of adapters and a list of domains on the HMC

– this determines the APQNs available to the compute nodes

Each (KVM or z/VM) guest running a compute node

– must be assigned a list of passthrough (aka dedicated) adapters and a list of domains

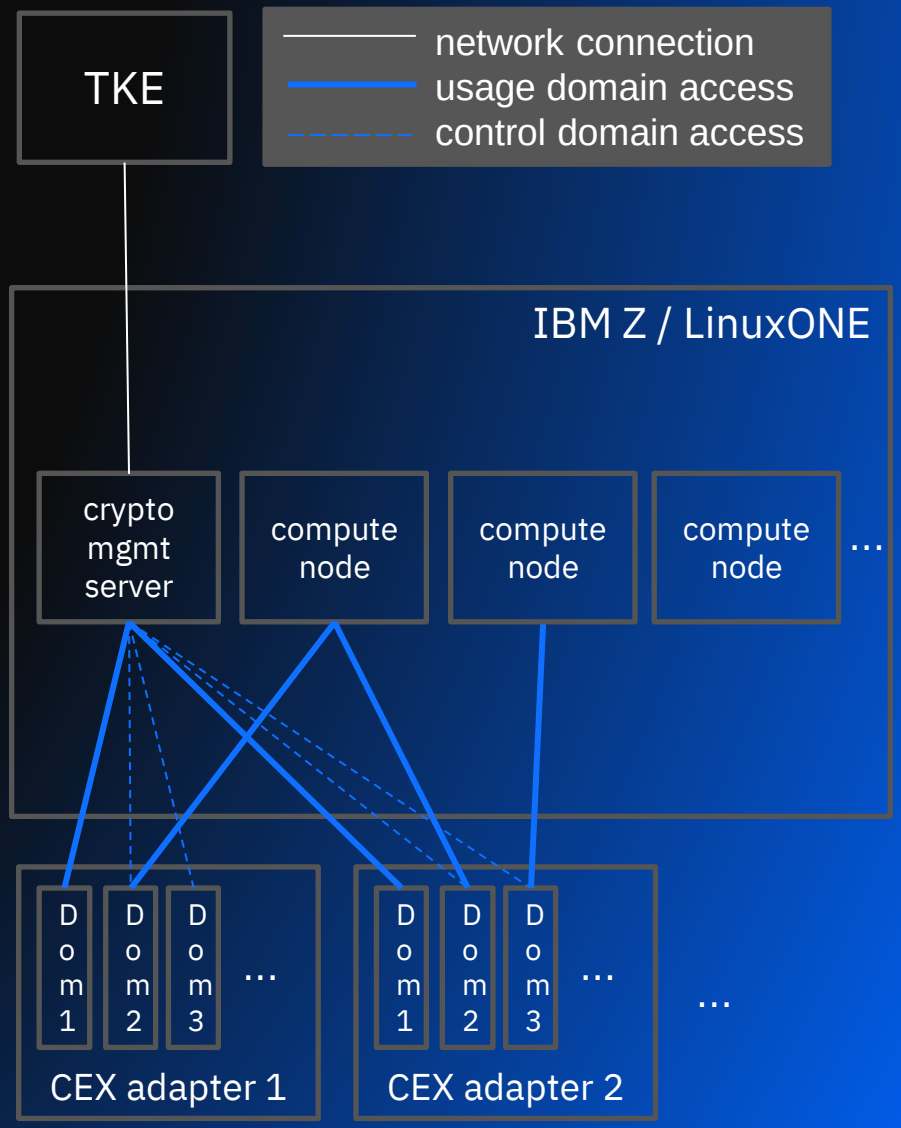
  • z/VM guests: in user directory

  • KVM guests: by assigning a mdev device with adapters and domains assigned

– this determines the APQNs available to the compute nodes

Managing CEX resources

- Management task:
  - defining administrators of adapter domains (take ownership of adapter domain)
  - setting master keys in adapter domains
- Performed using the TKE
  - to manage CEX adapters on a system the TKE must connect to a crypto management server on that system
- Crypto management server
  - must run a TKE daemon (CCA: catcher, EP11: ep11TKEd)
    - must be reachable by TKE via TCP/IP, using specific ports
  - must have access to one APQN for each CEX adapter to be managed
  - must have control domain access to each domain to be managed
  - a crypto management server can run in an LPAR or guest outside of RHOCP cluster
    - For example, Linux on Z in LPAR, KVM host, z/OS



Using CEX resources in K8s orchestrated containers

A ***CEX resource*** is a single domain within a CEX adapter

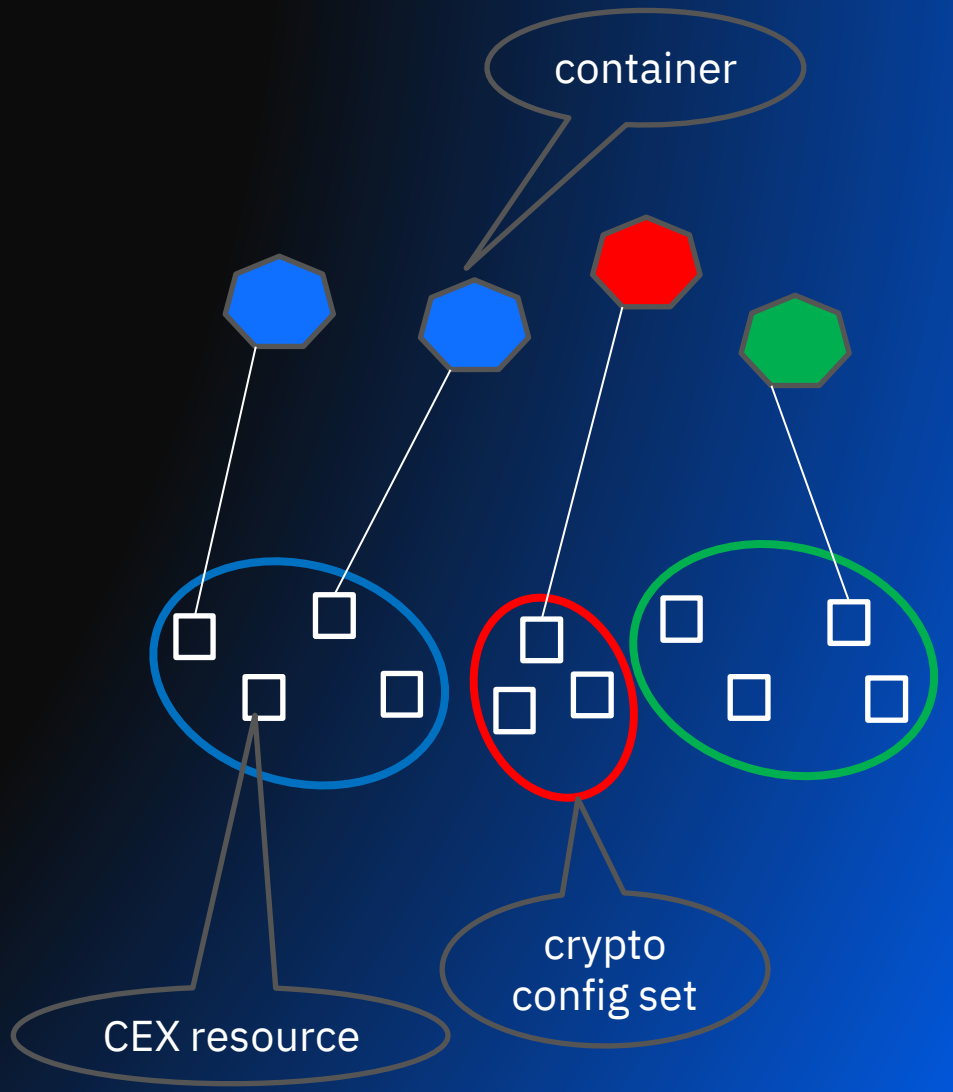
CEX resources are identified by their host, and their APQN (the pair of an adapter ID and a domain ID)

- have additional attributes like adapter mode (accel, CCA, EP11), or CEX generation (CEX5S, CEX6S, CEX7S)

Equivalent CEX resources can be configured to belong to a *crypto config set*

In version 1.0

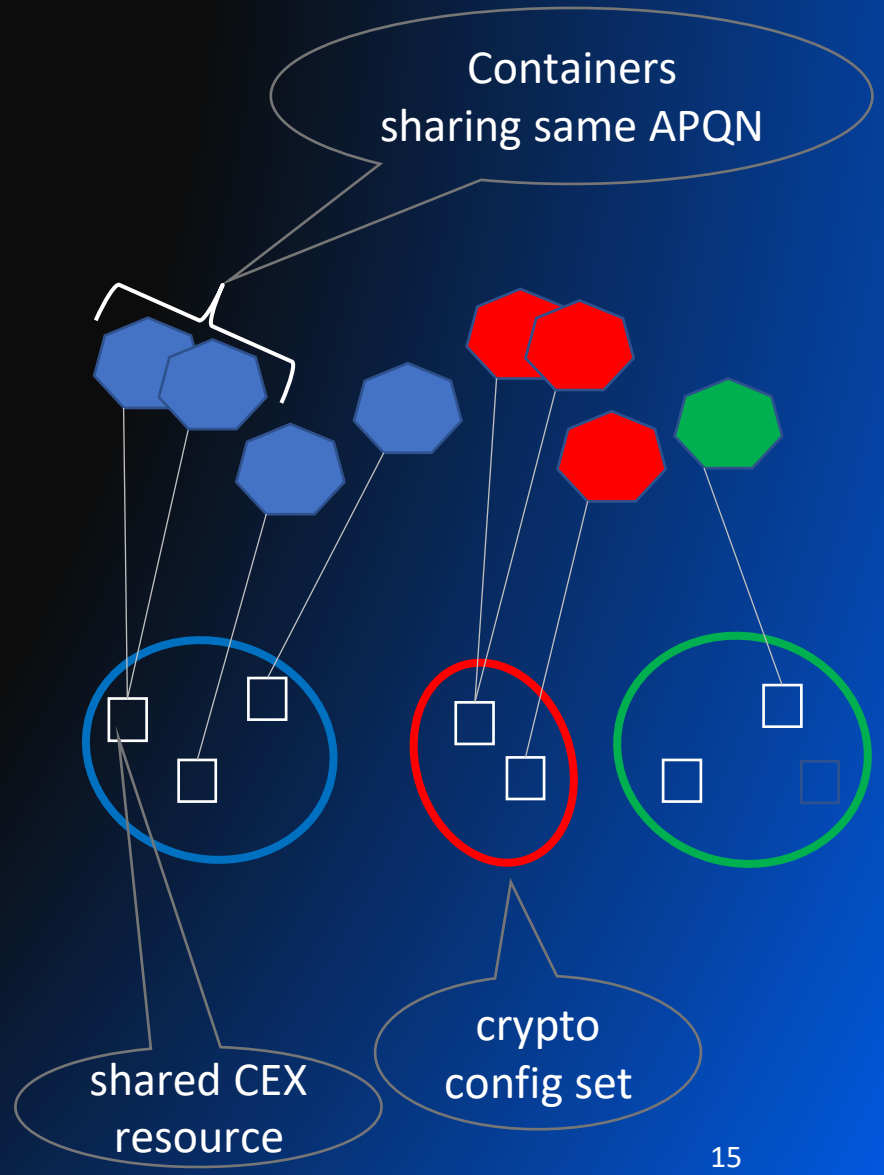
- each container can be assigned a single resource from a specific crypto config set
  - Note: a pod comprising multiple containers can use multiple CEX resources: one by each of its containers
- a crypto resource can be assigned to n ≥ 1 containers if the APQN overcommit limit n is configured in the deployment



Sharing CEX Resources

The cluster environment variable APQN_OVERCOMMIT_LIMIT determines how many containers may access (and thus share) a single CEX resource.

By default, APQN_OVERCOMMIT_LIMIT is set to 1.



Crypto config sets

The crypto config sets are defined in a config map.

V 1.0: each config set comprises

- set name
- project
- optional: a CEX mode
- optional: a minimal CEX generation
  - z.B. `mincexgen: "cex6"`
- a list of APQNs (+ optional machine IDs)

At any point in time a CEX resource must only be a member of a single crypto config set

- V1.0: no two crypto config sets may have the same APQN in their APQN list

The config map defining the crypto config sets can be changed at any time.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cex-resources-config
  namespace: kube-system
data:
  cex_resources.json: |
    {
    "cryptoconfigsets":
    [
     {
      "setname": "CCA_for_customer_1",
      "project": "customer-1",
      "cexmode": "cca",
      "apqns":
      [
        {
          "adapter": 1,
          "domain": 6,
          "machineid": ""
        },
        {
          "adapter": 2,
          "domain": 6,
          "machineid": ""
        },
        ...
      ]
     },
     ...
```

How can your containerized application take advantage of CEX resources?

Container deployment can request a CEX resource from a crypto config set with a resources statement:

```
...
spec:
  containers:
  - image ...
    ...
    resources:
      limits:
        cex.s390.ibm.com/red_cex_config_set: 1
...
```

Resource type CEX resource

Name of crypto config set

Must be 1 – only one CEX resource per container allowed

- At container start a compute node will be determined that has a free CEX resource from the specified crypto config set and the container will be assigned one of the CEX resources from the crypto config set.
- The CEX resource is chosen according to the config map valid at container start time.
- Later changes to the crypto config map do not affect running containers with CEX resources.
- When a container stops, its CEX resource is released.

What do CEX resources within a container look like?

Device node: `/dev/z90crypt`

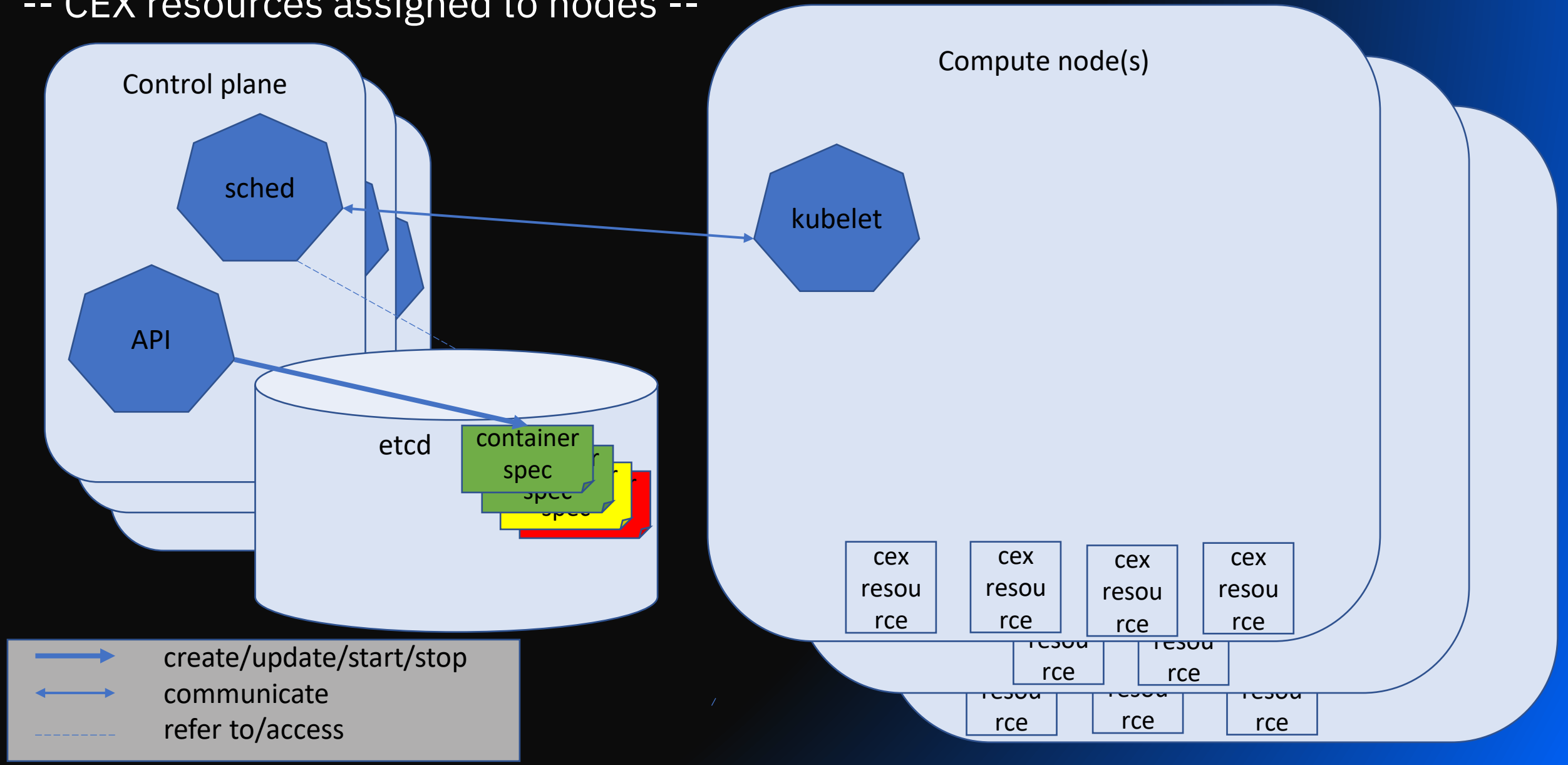
- Accessible
- Permits access to the assigned APQN only

Directories: `/sys/bus/ap` **and** `/sys/devices/ap`

- available read-only in container
- usable by lszcrypt (and chzcrypt)
- static: show status at container start time
- restricted to APQN assigned to container
- counters and load attributes are set to 0

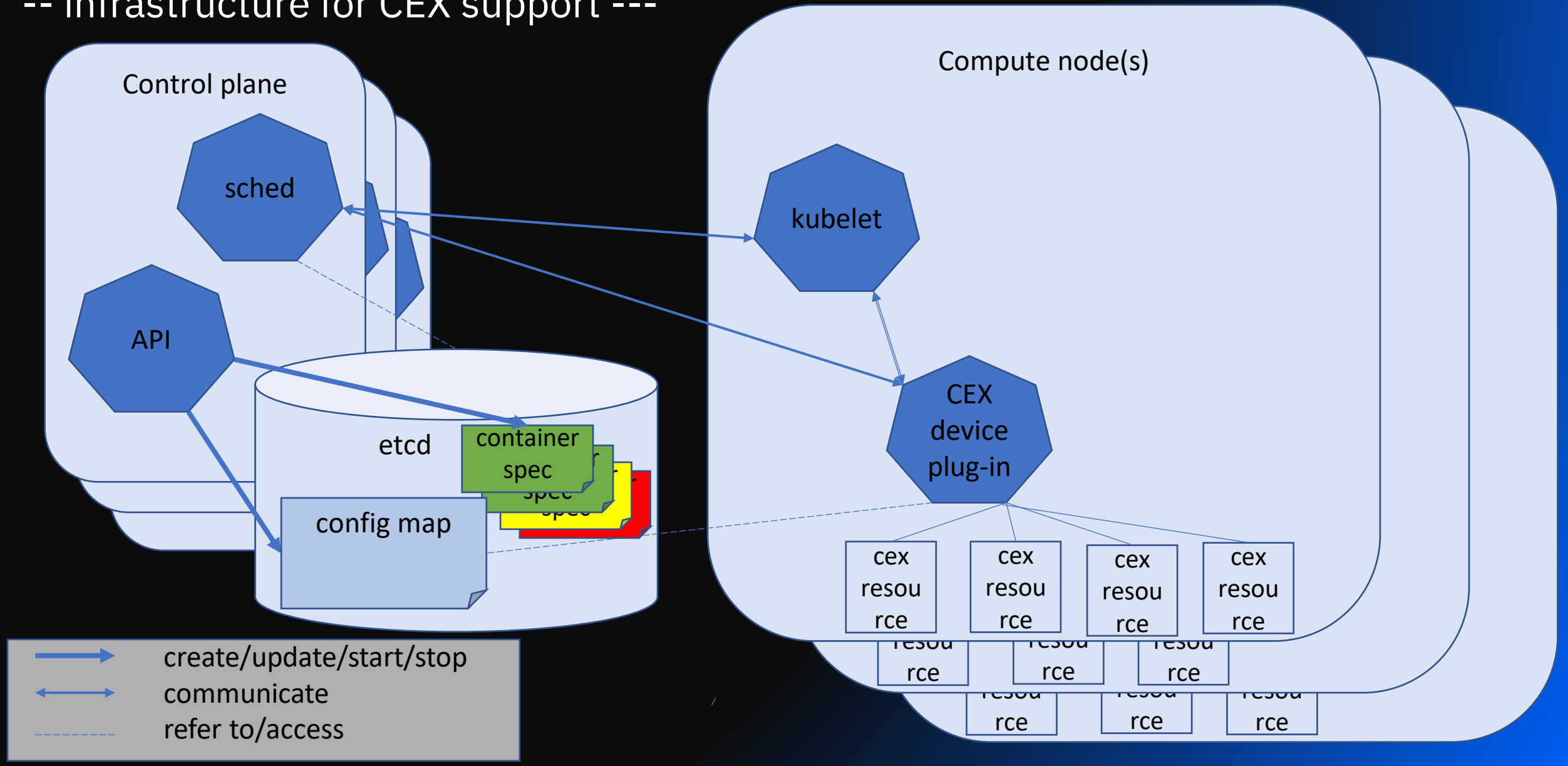
Kubernetes components for CEX support

-- CEX resources assigned to nodes --



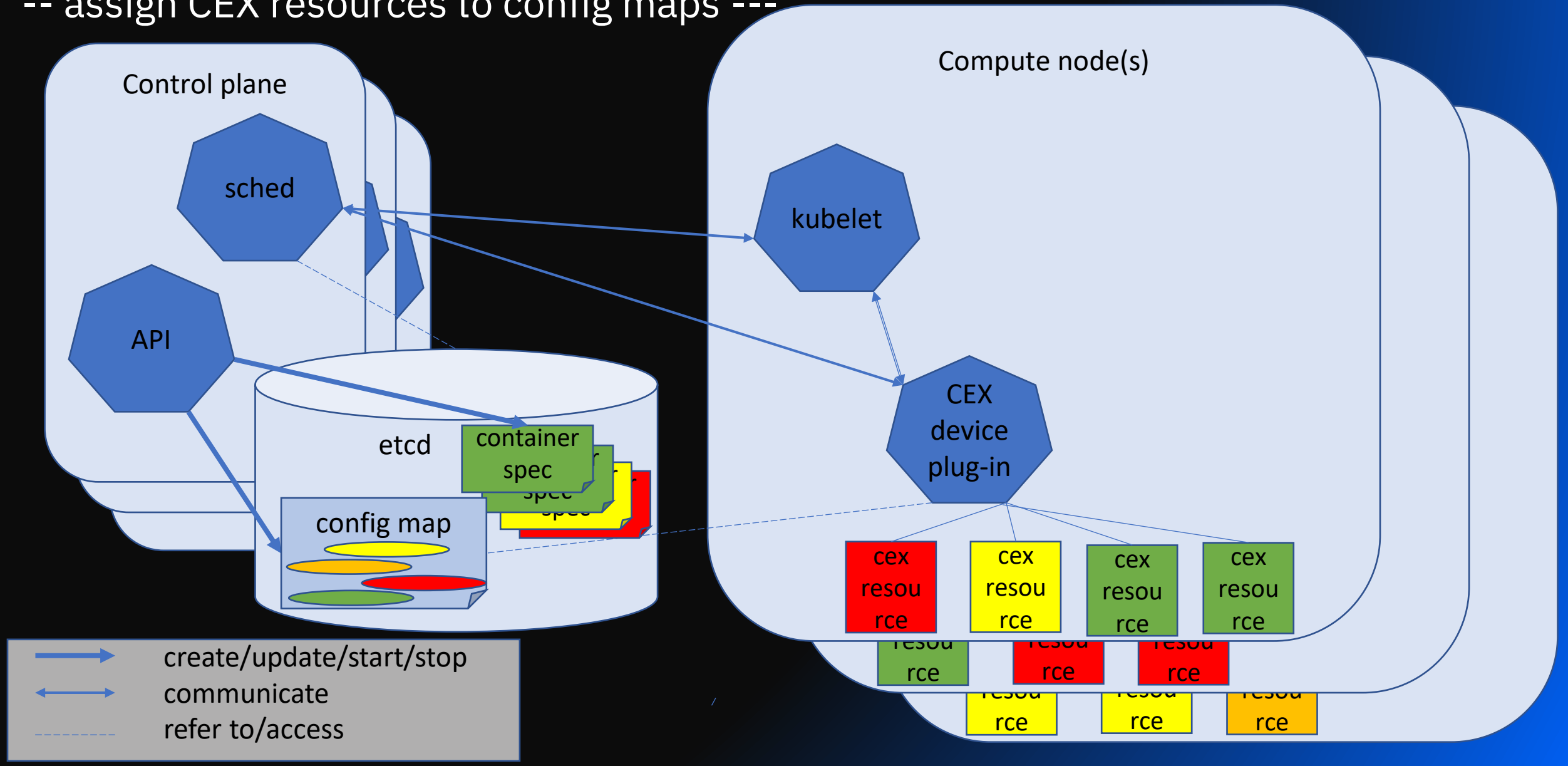
Kubernetes components for CEX support

-- infrastructure for CEX support ---



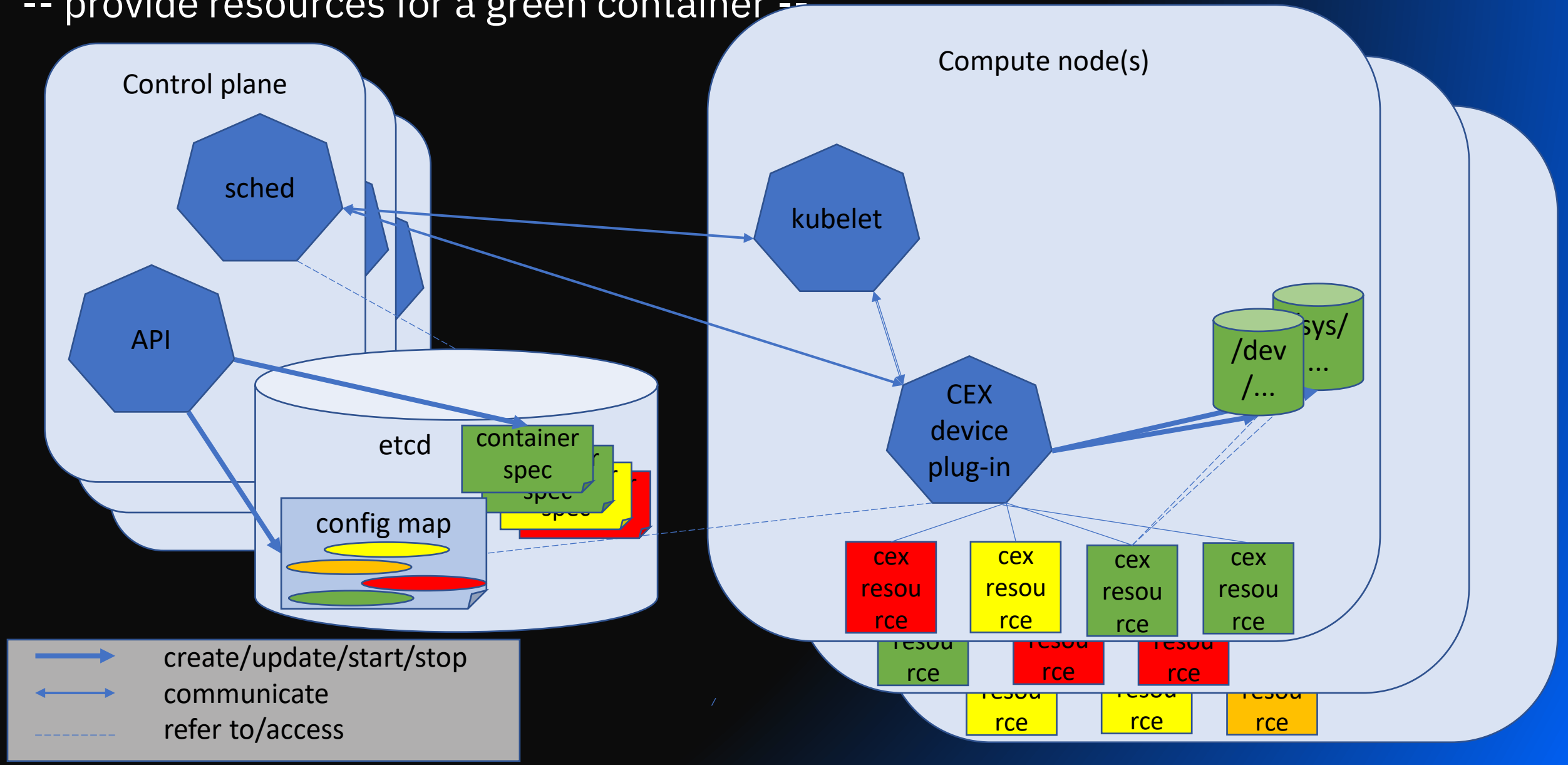
Kubernetes components for CEX support

-- assign CEX resources to config maps ---



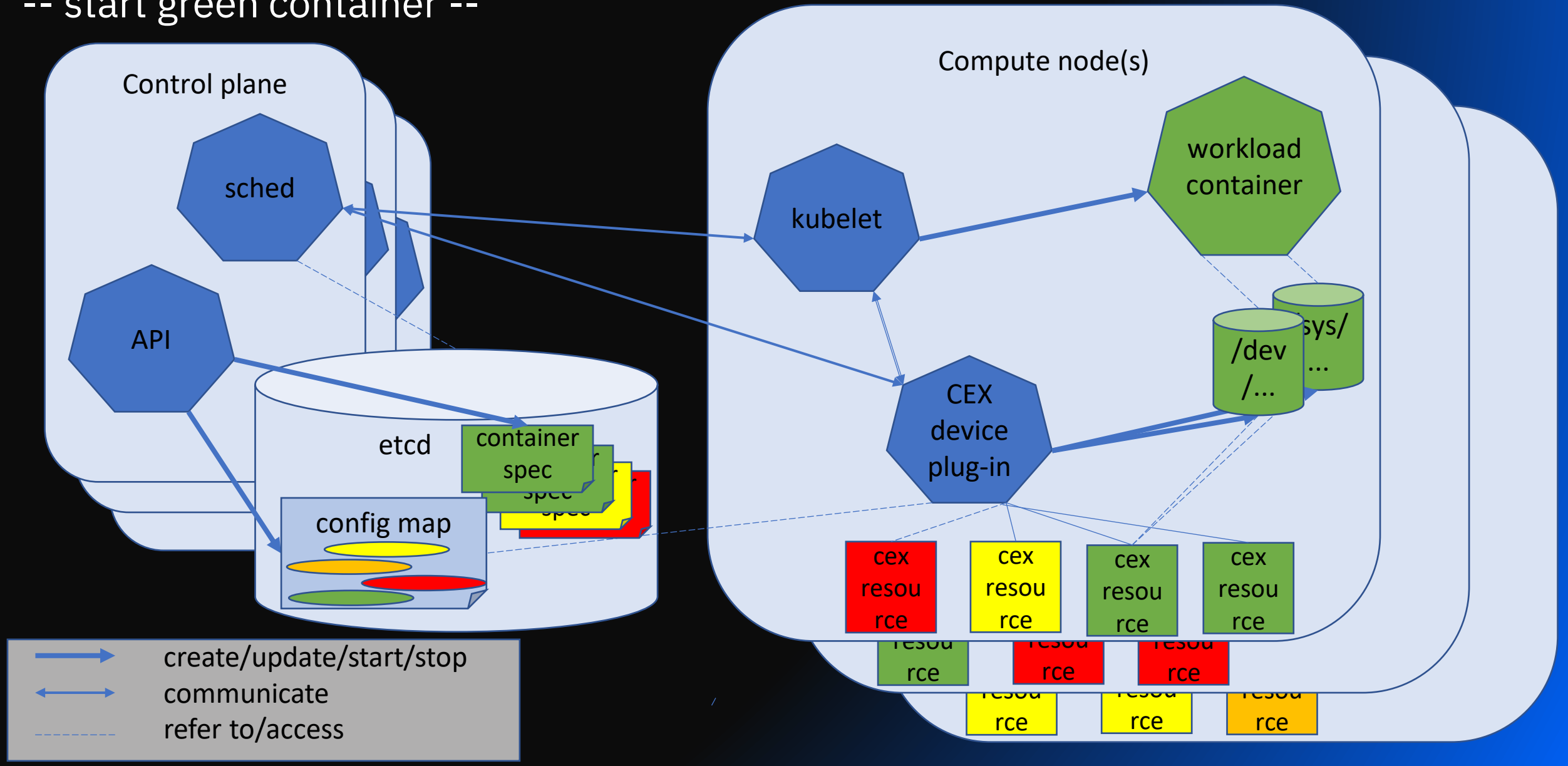
Kubernetes components for CEX support

-- provide resources for a green container --

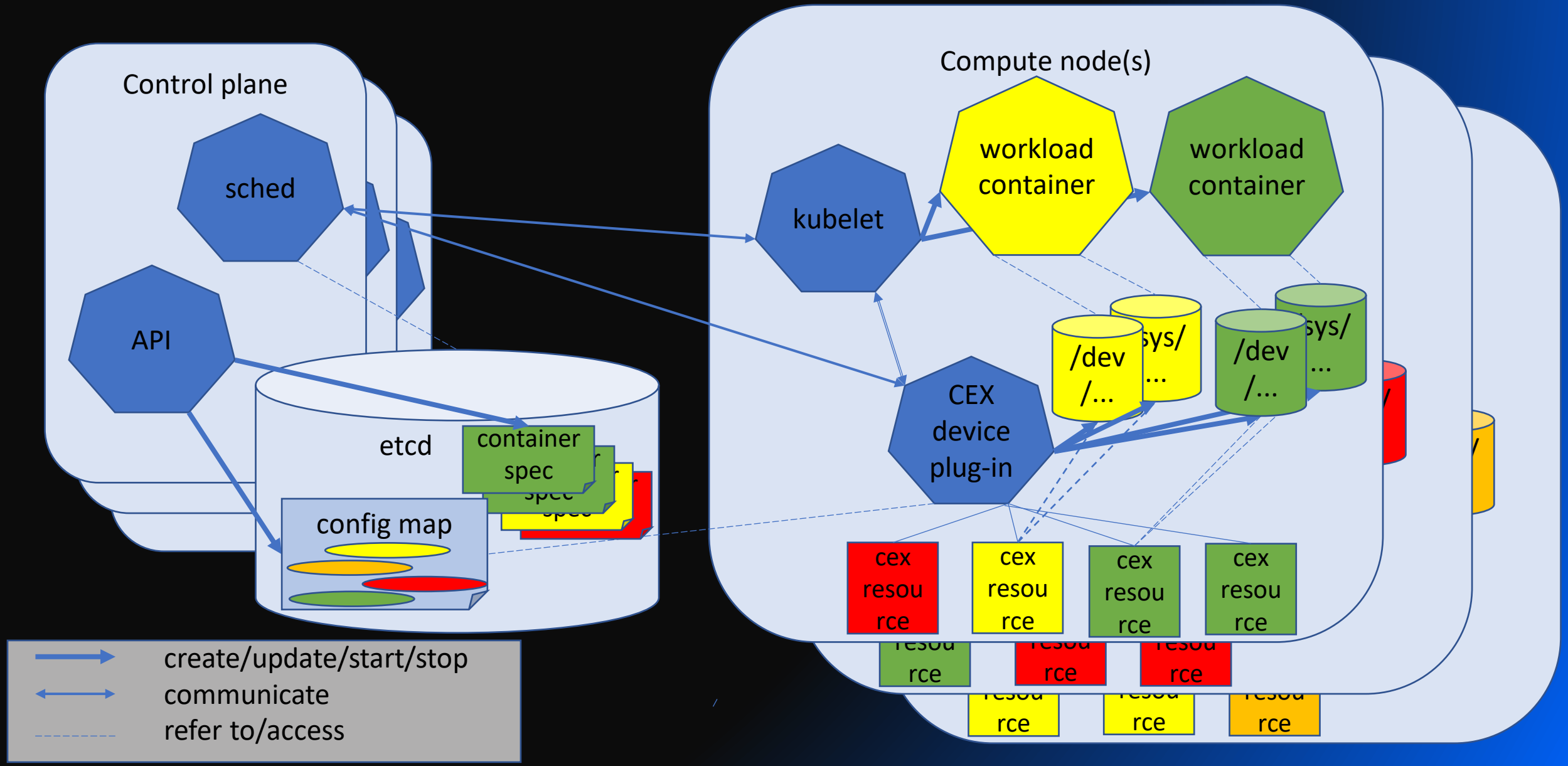


Kubernetes components for CEX support

-- start green container --



Kubernetes components for CEX support



The CEX device plug-in

- A privileged container (daemon set) running on each compute node.
- Communicates with the kubelet.
- Accesses current config map and detects CEX resources available on each compute node.
- When starting a new container with CEX resources from a crypto config set
  - detects appropriate CEX resources available on the node
  - generates a crypto device node to access this crypto resource (zcryptctl) to be accessed by new container
  - generates a shadow sys file system to be mounted into new container
  - assigns the CEX resource via kubelet device plugin API to the container
- After container termination, frees device nodes and shadow sys file systems no longer needed.

Restriction on project-based access control

Only containers belonging to a project that is specified in the project attribute of the crypto config set shall access CEX resources from that crypto config set.

- Restriction in version 1.0
  - The CEX plug-in does not enforce that the project attribute of the crypto config set matches the project of the container to which a CEX resource shall be assigned.
  - The CEX plug-in detects a mismatch of the project attribute of the crypto config set, and the project of a running container and logs such mismatches.
- Mitigation
  - For each project, set the quota for a crypto config set to zero unless the project is specified in the config set.
  - There is a script to support the generation of quotas for new projects.

What needs to be considered for using the CEX device plug-in?

Planning your RHOCP environment

- Assign CEX adapters and domains to compute nodes (LPAR and hypervisor level).
- Run at least one crypto management server connected with the TKE on each system.
  - The crypto management servers must have access to all control domains for which the associated usage domains shall be used by the RHOCP cluster
  - The crypto management servers must have access to all adapters (and one usage domain) which shall be used by the RHOCP cluster.
- Ensure equally configured APQNs are distributed over multiple nodes to enable flexible container placement, workload migration, and high availability.
- Define a config map with crypto config sets
  - grouping equally configured APQNs into the same crypto config sets.
- Define containers requesting CEX resources from crypto config sets.

How to get started with the CEX device plug-in?

Identify your CEX resources and create crypto config sets

- Define your crypto config sets and create the `my-cex-resource-config.yaml`
- Create the config map, for example, with

```
kubectl create -f my-cex-resource-config.yaml
```

Install the K8s device plug-in for IBM Crypto Express

- Download `deploy_cex_plugin_daemonset.yaml` from GitHub or create your own
- Modify the yaml file to use community (default) or RH certified container image for RHOCP
- Install the device plug-in with

```
kubectl create -f deploy_cex_plugin_daemonset.yaml
```

- Verify the installation, for example, with `kubectl get pods -n kube-system`

Use cases

Acceleration of RSA

SW using openSSL + ibmca engine

- e.g., Apache, OpenSSH

Java SW using IBMPKCS11Impl + openCryptoki w/ icatoken

- e.g., IBM WAS

HSM protection of most valuable keys for SW with PKCS #11 plugin

protect master key of key repository with HSM

– e.g., GKLM, Postgres

Protect signing key of TLS server

- e.g., Apache

Custom workloads using HSM based crypto libraries

fincanial SW using CCA host library

SW with PKCS #11 APIs (-> openCryptoki)

Blockchain SW using grep11

Kernel-based crypto

cryptography cannot be HSM-protected in a way that is specific for each container.

- e.g., dm-crypt, IPSec

Protected keys are valid in all containers running inside the same node.

Release, Support, References

Release

- Community container images are published on quay.io
  - `quay.io/ibm/ibm-cex-plugin-cm:v1.0.0`
- Certified image in the RH catalog for use with RHOCP
  - IBM CEX Device Plugin CM

Support

- Support of community container images via Github Issues and Pull Requests
- Support for the Red Hat certified container by opening a ticket with Red Hat (typical support flow for Red Hat certified containers)

References

- Sources https://github.com/ibm-s390-cloud/k8s-cex-dev-plugin
- Documentation https://github.com/ibm-s390-cloud/k8s-cex-dev-plugin/docs/docu.md

  https://www.ibm.com/docs/en/linux-on-systems?topic=openshift-kubernetes-crypto-plug-in

Call to action!

Please provide feed back

- What works well?
- What is missing?
- Which requirements are not fulfilled?

Let us know

- What kind of crypto workload do you want to bring to RHOCP?
- Are you interested in a Design Thinking Workshop for crypto usage in RHOCP?

Speak up or send an e-mail to

Reinhard Bündgen buendgen@de.ibm.com
Hendrik Brückner brueckner@de.ibm.com

Questions?