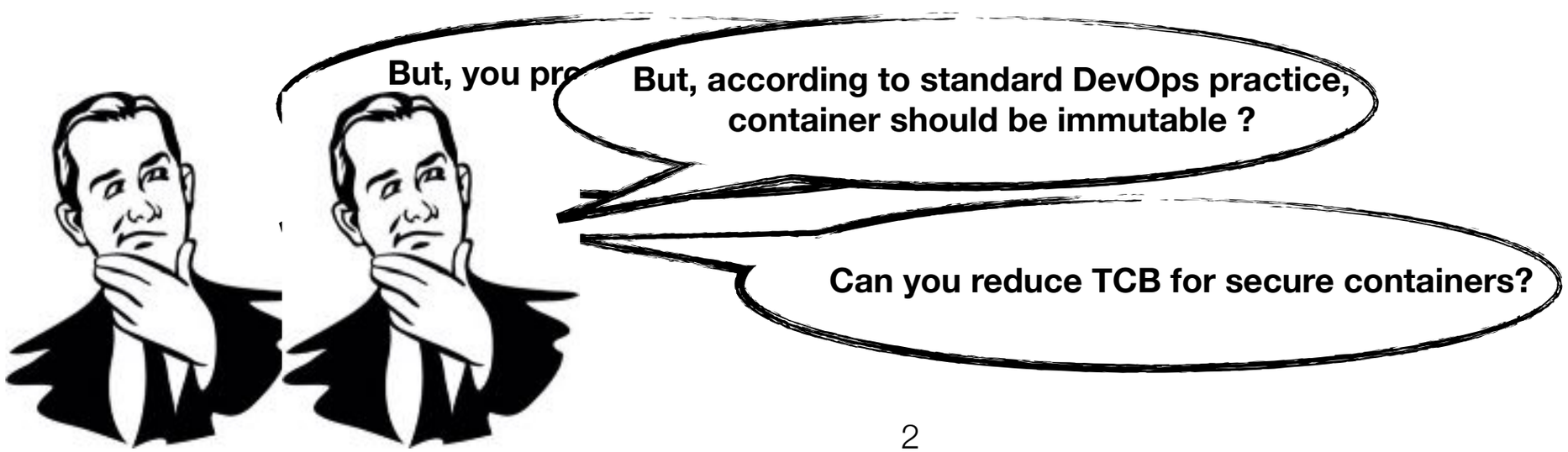
RECap: RunEscape Capsule for On-demand Managed Service Delivery in the Cloud

Shripad J Nadgowda, Sahil Suneja, Canturk Isci

IBM T J Watson Research Center

Evolution of application runtimes
(General-purpose —> Specialized)



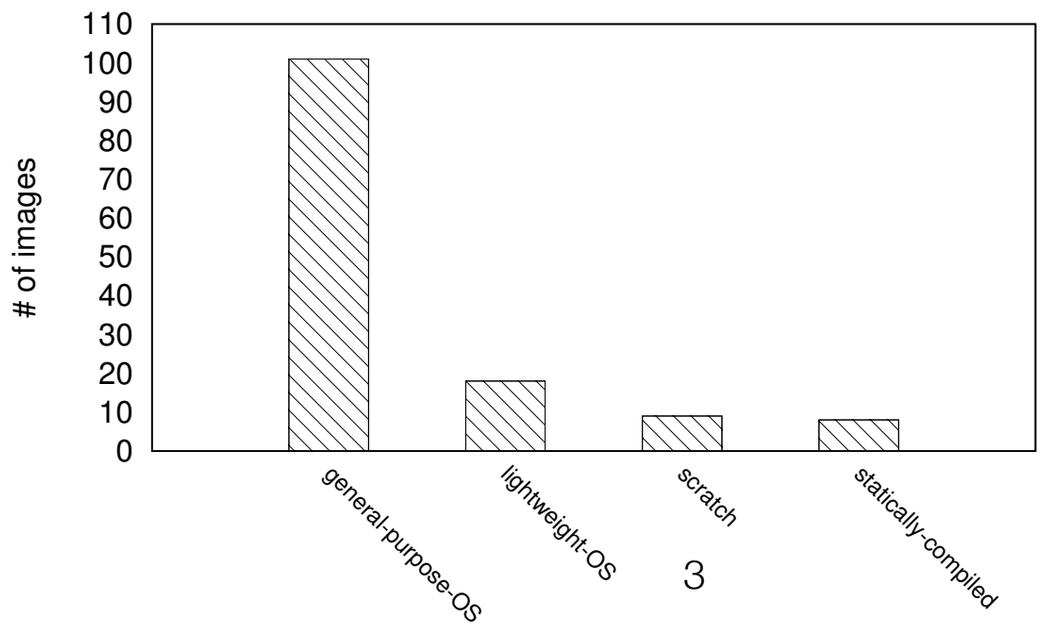
Evolution of application runtimes
(General-purpose —> Specialized)

Physical Machines/ VMs

Containers

Lightweight runtimes

Microcontainers



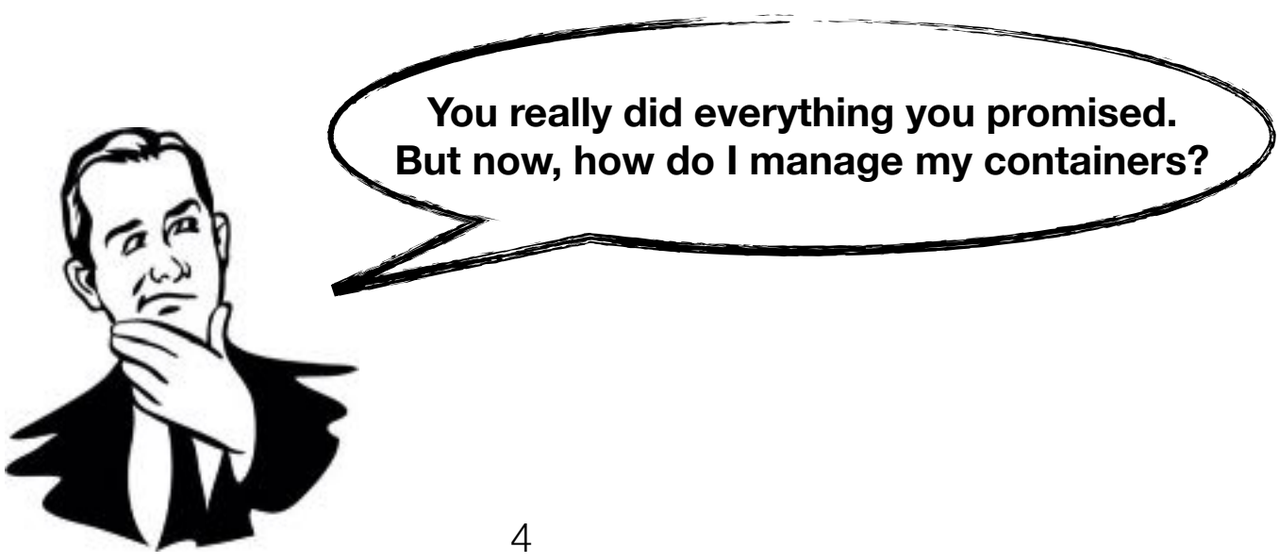
Evolution of application runtimes
(General-purpose —> Specialized)

Physical Machines/ VMs

Containers

Lightweight runtimes

Microcontainers



3 Rules for Modern Container Cloud

Rule 1

Functionality Dis-aggregation: **Break your traditional monolithic applications into two parts, namely— core application functions and other auxiliary functions**

Rule 2

Use Micro-containers: **Package and run your core application functions through micro-containers for safer execution**

Rule 3

On-demand Auxiliary functions: **Enable delivery common auxiliary functions as on-demand managed services on cloud**

What are these auxiliary functions?

System and Application Administration

Typical cronjobs:
malware scans,
logrations,
ntpupdate

Application Utilities:
MD5 checksum,
archival with tar

Debugging

Ad-hoc debugging:
gdb,
strace,
tcpdump,
iperf

Monitoring

System metrics:
CPU,
Memory

Application Metrics:
# of connections,
# of requests

How to deliver these functions on-demand and securely to running containers ?

Inspiration…

Serverless
or
Function-as-a-Service

Introducing RunEscape Capsule (RECap)

(also stands for Capability Redemption)

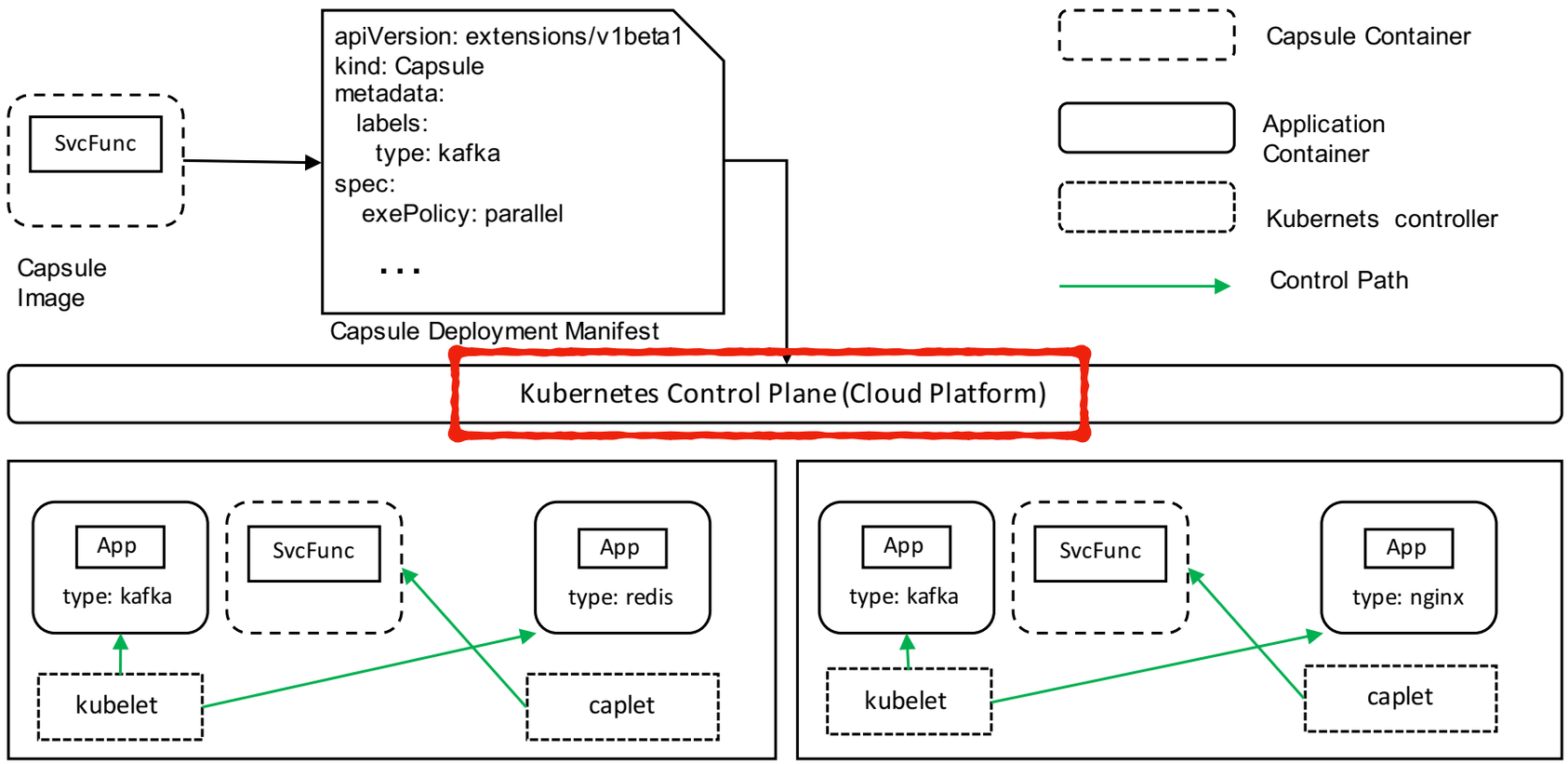
Capsule

- This itself is a micro-container
- Encapsulates auxiliary function and all its dependencies together
- Existing tools and techniques (e.g. Dockerfiles) can be leveraged to create an image

RunEscape

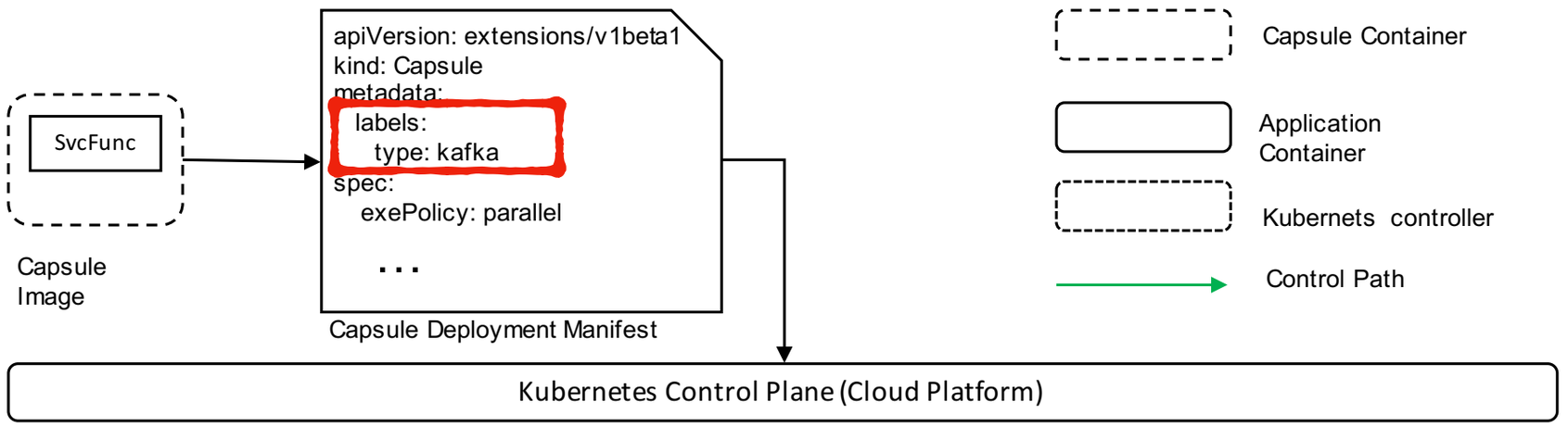
- Capsule are dynamically-attached to app containers as a sidecar container
- Capsule is attached ONLY for the duration of running the function
- Capsule is then detached or Escape the app container

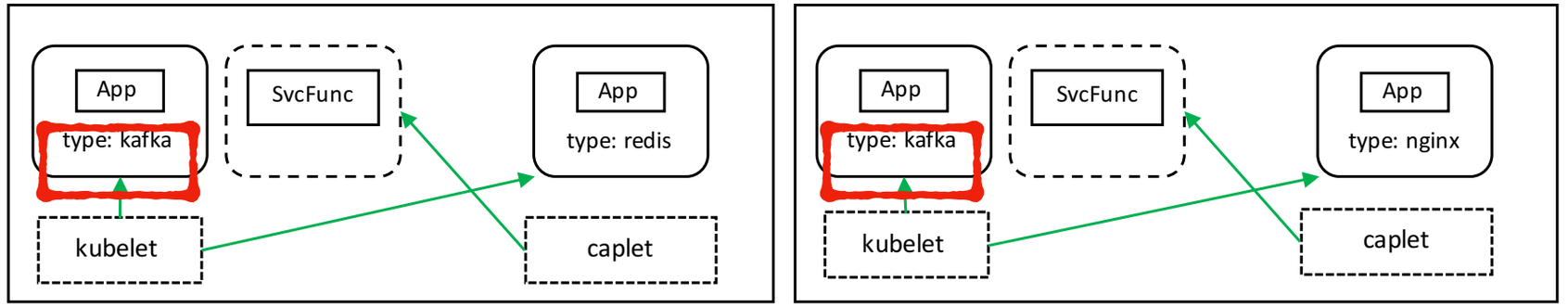
RECap: System Design



Although agnostic to the underlying cloud substrate, we are currently designing RECap for Kubernetes

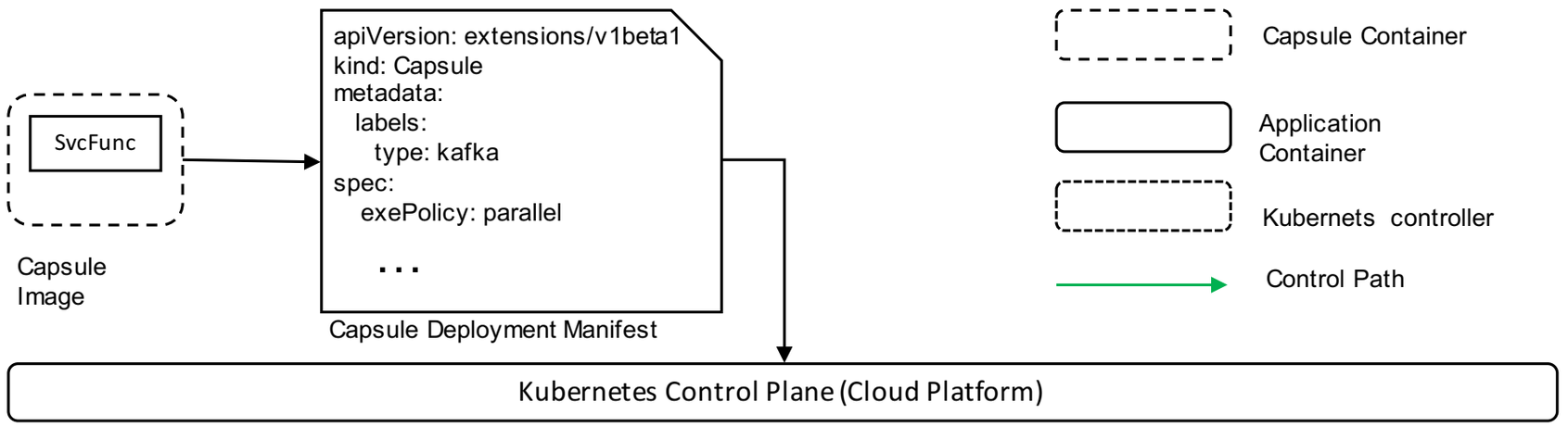
RECap: System Design

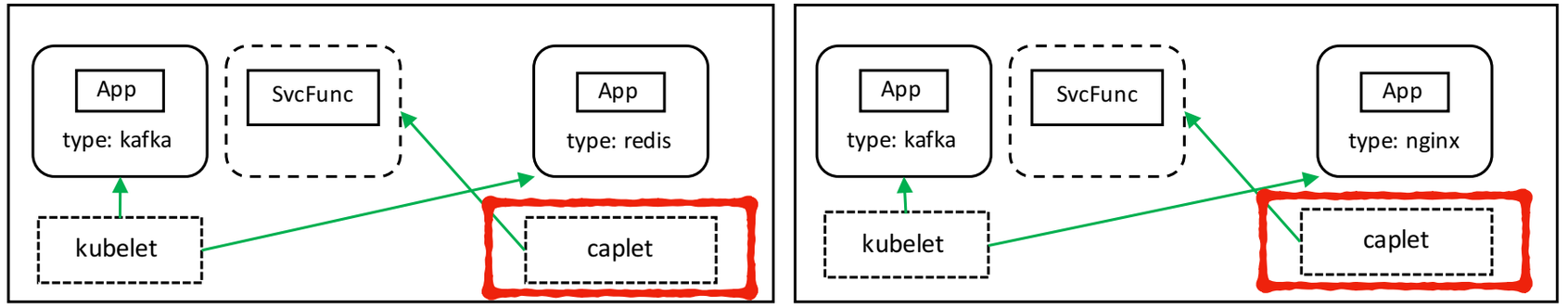




Affinity between capsule and app container is established through K8s labels

RECap: System Design





A special node agent "caplet" manages lifecycle of capsule containers

RECap: Evaluation

	Image Build	App Deploy	Docker Exec	Capsule RunEscape
Function exists in app container	0	0	0.083s	0
Function does not exist in app container	5.24s	0.29s	0.083s	0
Capsule Image is present on the node	0	0	0	0.243s
Capsule Image is not present on the node	6.2s	0	0	0.243s

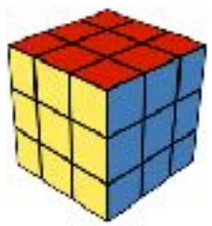
Recap on RECap

Trend

Breaking monolithic applications into Microservices and delivering them in Microcontainers

Challenge

Ability to implement common system administration, debugging, monitoring functions on-demand



Solution

RunEscape Capsule is a framework that promotes a cloud-native solution for on-demand managed service delivery

RECap: Discussion

What is the criteria for deciding which functions can be de-coupled from application and delivered through RECap ?

Is it safe to dynamically execute on-demand functions in the application context ?

Whom does RECap is really going to help ?
Developer, Administrator, Cloud provider, Everyone ?

Thank You

nadgowda@us.ibm.com