



Real Containers, Digital Containers

CLdN's Red Hat OpenShift Journey

Bart Coucke, Head of Operations, CLdN IT Systems

Red Hat Summit Connect Brussels · 15 October 2026

CLdN at a glance

A leader in integrated maritime links

30 +

ships

200 +

sailings per week

9,000 +

multimodal units

3,000

colleagues across Europe



1

The challenge



10 years of the same model

Where we started

- ◆ **One day a month.**
A classic support contract: problems were collected, an engineer came by & we called when something was on fire.
- ◆ **Software reaching end-of-life.**
Libraries to update, full clients to retire, a push toward web & containers.
- ◆ **New projects, no room for trial and error.**
A first production platform was needed in weeks, not months.

The decision

“

Build it ourselves, or get it right from day one?

”

We chose not to experiment on production.





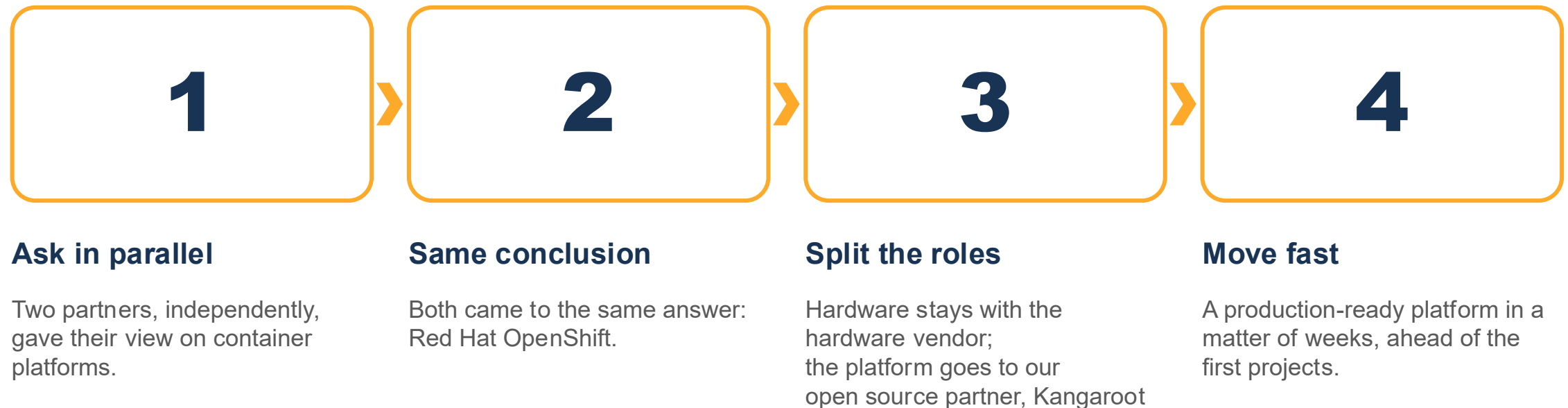
2

The choice



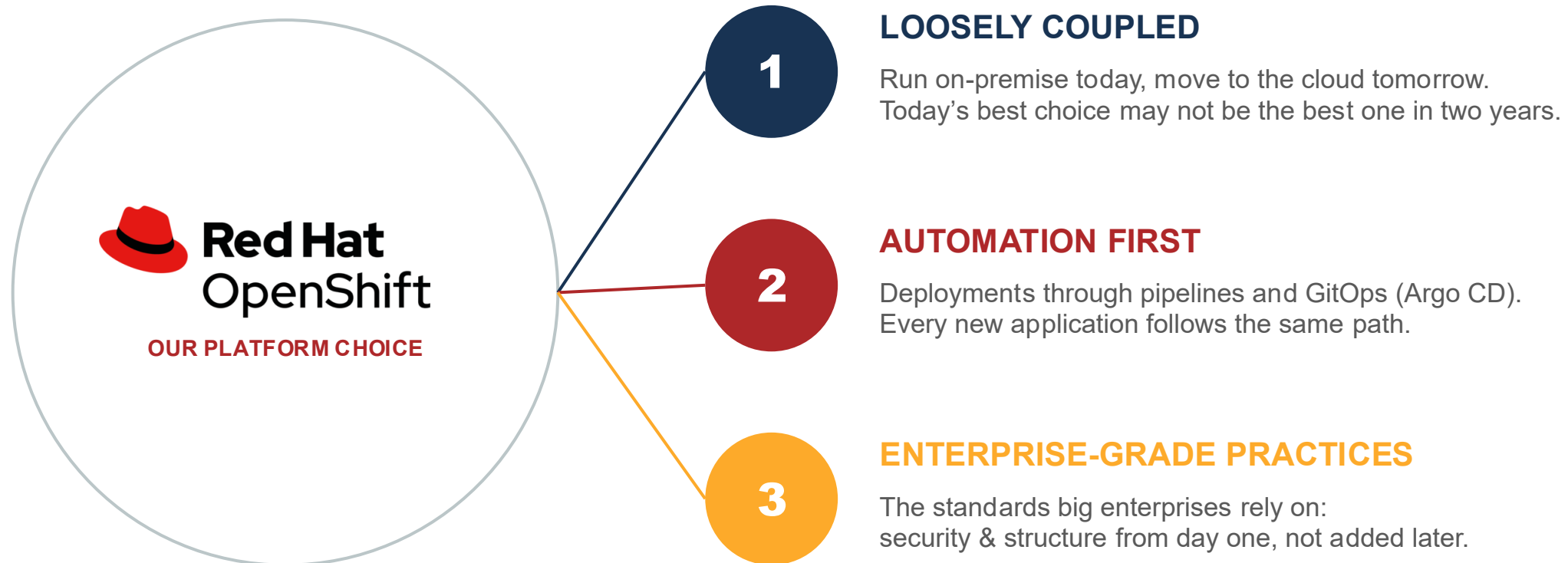
How we got to Red Hat OpenShift

The decision in four steps



Why OpenShift?

3 reasons that mattered



Three jobs, not one

Set up, run, guide: who does what

1 SET UP

Build the platform right

KANGAROOT

Designs & builds the platform, sets the standards.

CLDN

Defines the needs & the first applications.

2 RUN

Keep it healthy

KANGAROOT

Upgrades, patching, monitoring & 24/7 escalation.

CLDN

Runs the applications on the platform.

3 GUIDE

Help the team grow into it

KANGAROOT

Coaches the team on pipelines and deployment.

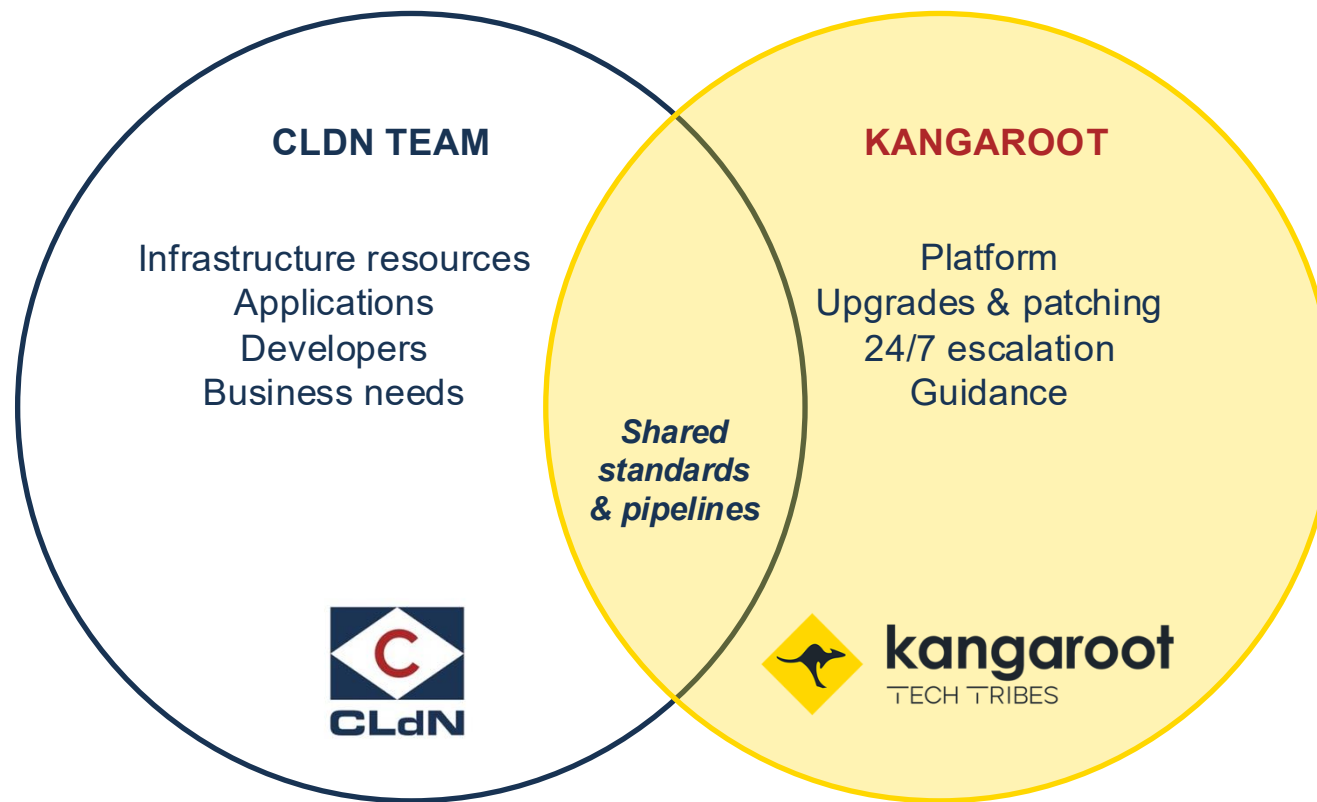
CLDN

Learns & takes on more over time.

Setting it up is one. Managing it is two. Guiding the team is the third.

1 team, 2 roles

How Kangaroot's Managed Services fit in



WHAT THAT LOOKS LIKE DAY TO DAY

- ◆ Fixed engineers who know the CLdN context
- ◆ A duty manager for escalations, day & night
- ◆ Planned upgrades, announced in advance
- ◆ A regular operations review



3

The journey

From first cluster to first apps

What happened & what we learned

Phase	What happened	What we learned
Platform	OpenShift cluster up & running, pipelines set up with guidance	Knowledge transfer matters as much as the build

First project

Delayed & eventually stopped

New technology takes time; plan for it

We tested the app. Not the failure.

Lesson 1: high availability

WHAT HAPPENED

- ◆ The first production application went live.
- ◆ When a node dropped, it did not survive.
- ◆ Not a platform problem:
pods were not spread across nodes &
failure had never been tested.

WHAT WE CHANGED

- ◆ High-availability testing is part of the delivery.
- ◆ Applications are written to tolerate one or two nodes disappearing.
- ◆ Upgrades are planned & happen without visible impact.

Standards first, freedom second

Lesson 2: structure from day one

A FRAME FOR DEVELOPERS

Instead of everyone deploying their own way, the platform defines how things are done.

A REAL LEARNING CURVE

Networking and storage concepts differ from VMs.
The first time is always the hardest.

A PLATFORM THAT STAYS CURRENT

A cluster that falls behind on versions ends up out of support.
We plan upgrades instead.

Where we go from here

Next steps, as far as we want to share

- ◆ More applications.
- ◆ A platform that stays current.
- ◆ The team grows into it.

Three things to take home

What we would tell anyone starting out

- 1** Don't pioneer alone
- 2** Test for failure, not just function
- 3** A platform needs setting up, running & guiding



Bart Coucke
Head of Operations, CLdN IT Systems

Questions?