KubeCon | CloudNativeCon

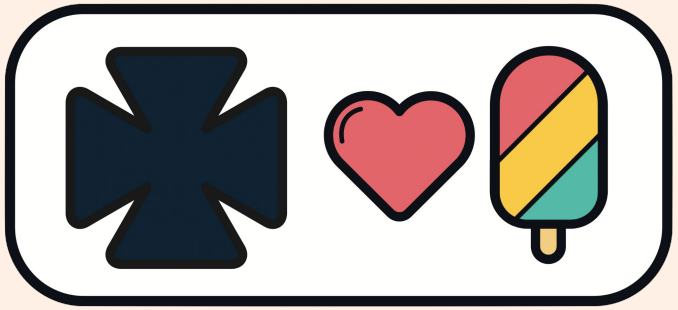
Europe 2022

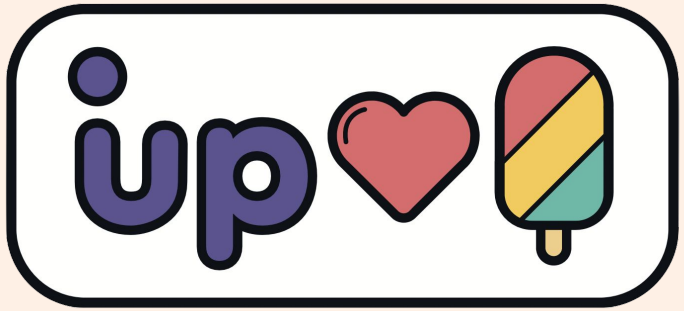
WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

Building Digital Twins For DFDS with Crossplane and Kubernetes

Tobias Andersen, DFDS – Matthias Lübken, Upbound



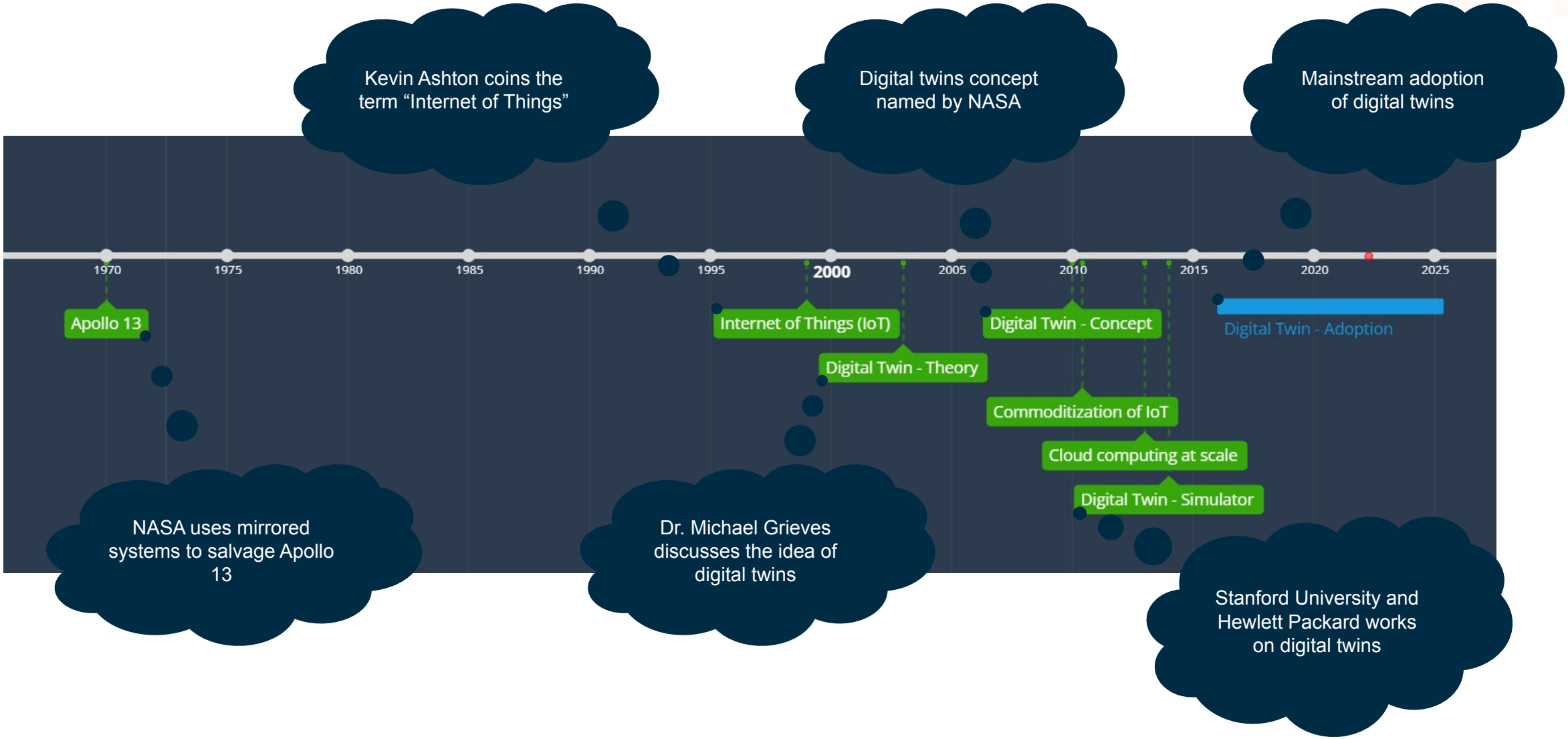


Agenda

- Big Picture
- Use-Cases
- Digital Twins
- Digital Twins with Crossplane
- Conclusion

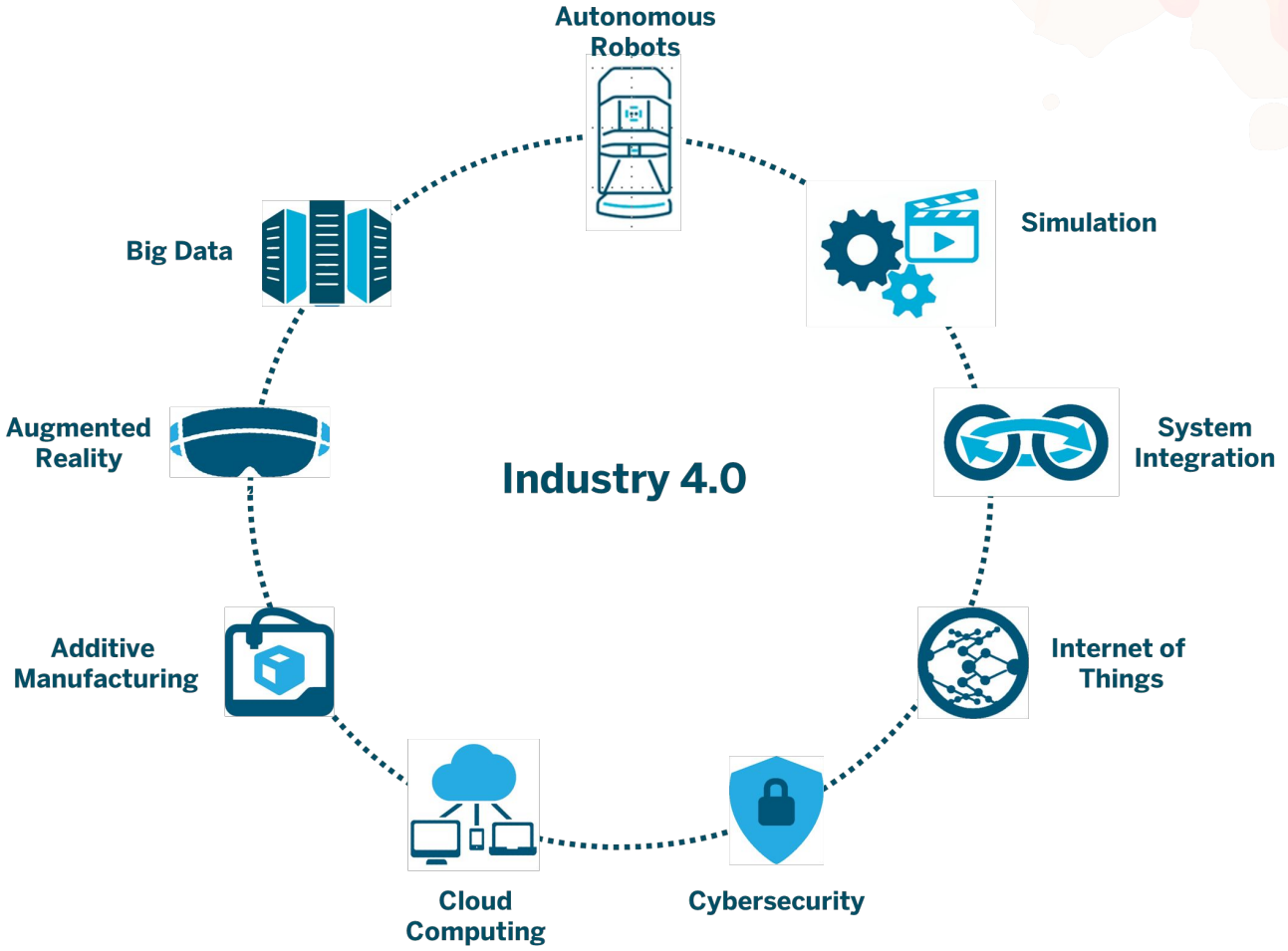
Big Picture

Big Picture

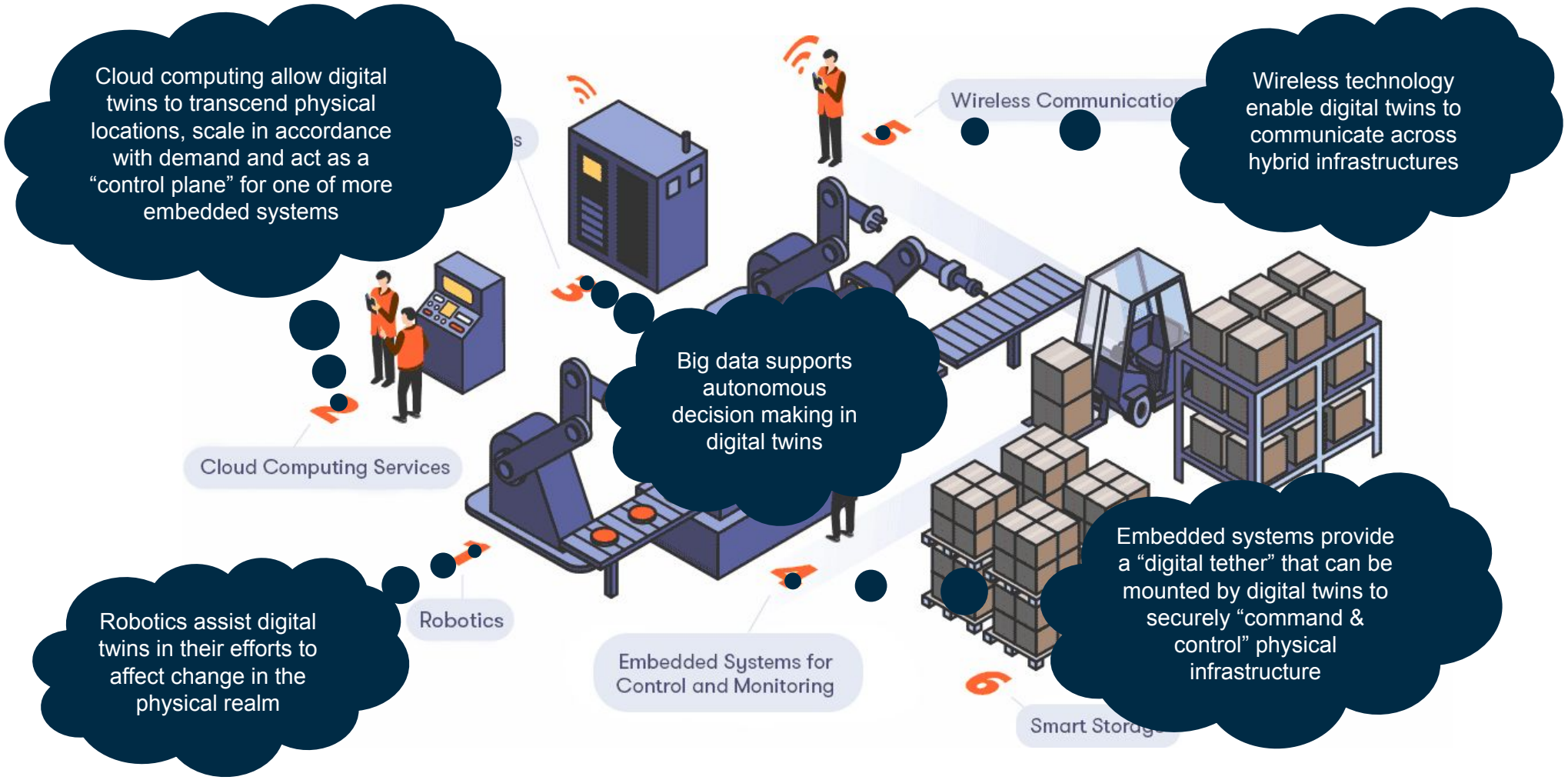

Kevin Ashton coins the term “Internet of Things”
Digital twins concept named by NASA
Mainstream adoption of digital twins
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
Apollo 13
Internet of Things (IoT)
Digital Twin - Concept
Digital Twin - Adoption
Digital Twin - Theory
Commoditization of IoT
Cloud computing at scale
Digital Twin - Simulator
NASA uses mirrored systems to salvage Apollo 13
Dr. Michael Grieves discusses the idea of digital twins
Stanford University and Hewlett Packard works on digital twins

Big Picture

- Digital twins are digital representations of physical assets
- Digital twins mimic behavioral models in physical systems
- Digital twins offers a digital control plane for a physical world
- Digital twins tethers to their physical counterpart from cradle to grave



Big Picture


Cloud computing allow digital twins to transcend physical locations, scale in accordance with demand and act as a “control plane” for one of more embedded systems
Wireless Communicatio
Wireless technology enable digital twins to communicate across hybrid infrastructures
Big data supports autonomous decision making in digital twins
Cloud Computing Services
Robotics assist digital twins in their efforts to affect change in the physical realm
Robotics
Embedded Systems for Control and Monitoring
Embedded systems provide a “digital tether” that can be mounted by digital twins to securely “command & control” physical infrastructure
Smart Storag

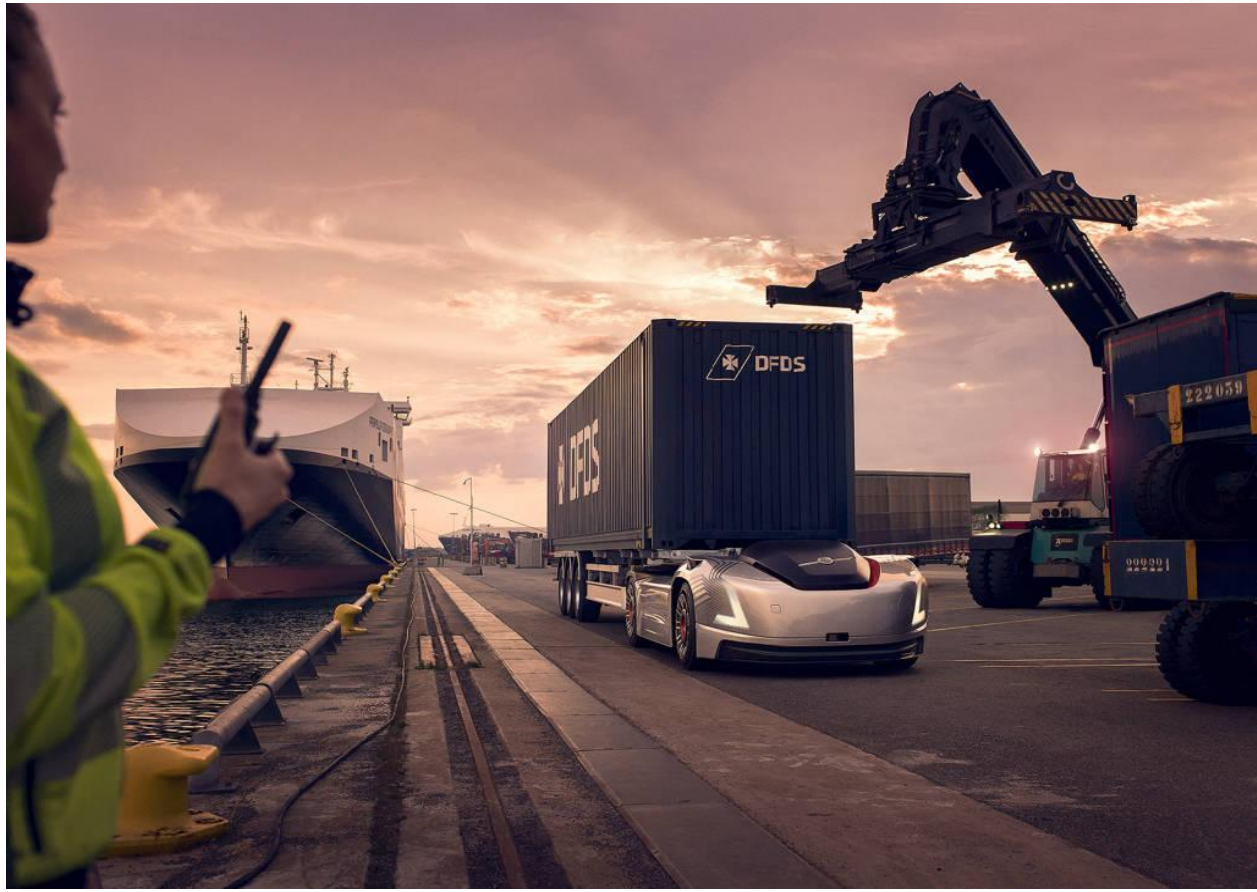
Big Picture

Big Picture

Use Cases

Use Cases > Trailers

- Reduce empty loads
  to minimize CO2-emissions
- Ensure integrity of “cold storage”
  services
- Support predictive maintenance
  on key assets
- Track goods
  in global supply chains



Use Cases > Shipments

- Improve routing
  to reduce fuel-consumption
- Centralizes management
  of IoT sensors
- Enables simulation
  of future shipments

Use Cases > Terminals

- Support automation of terminal processes
- Optimize space utilization in new facilities
- Increase productivity of personnel
- Leverage data to reduce energy consumption

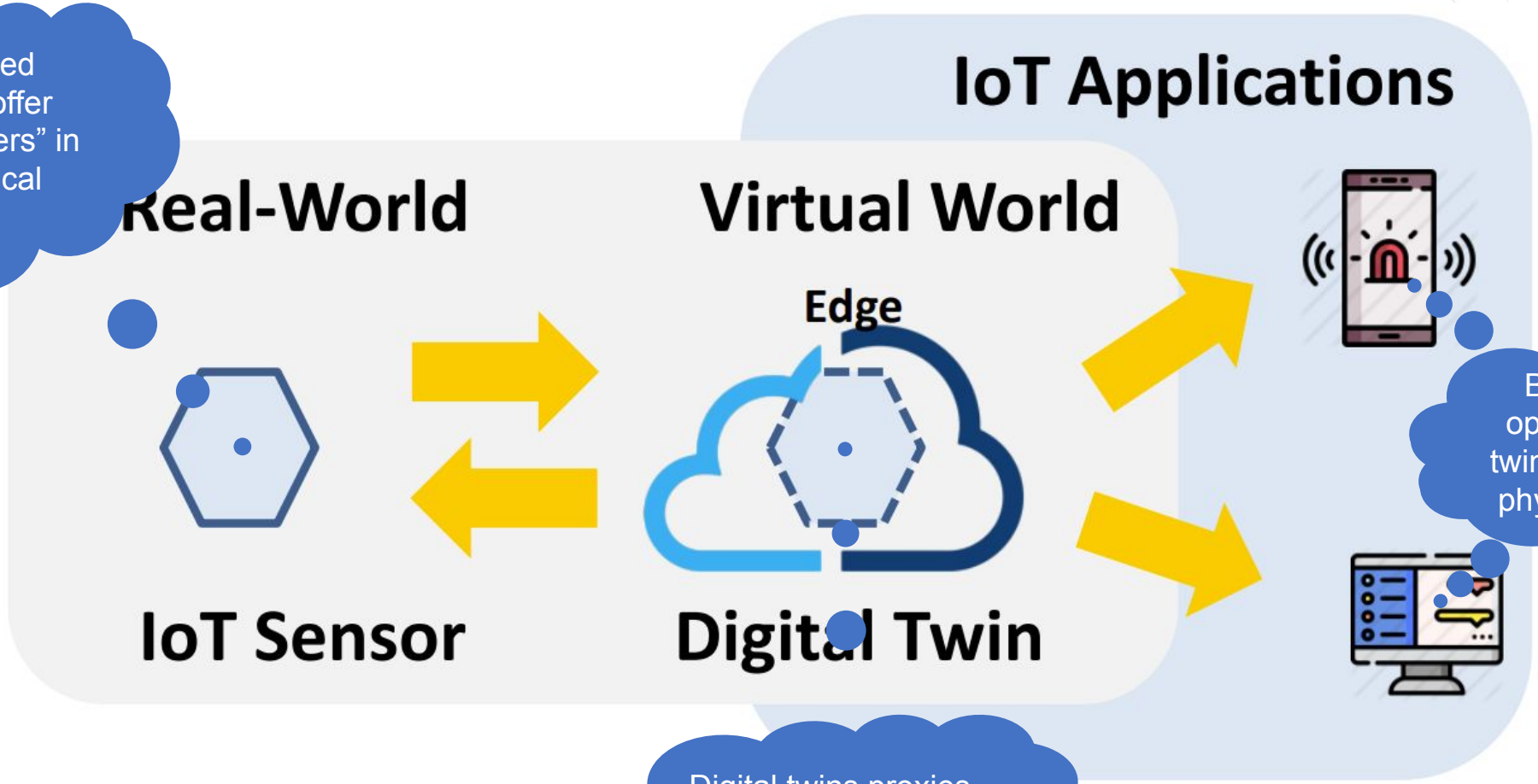
Use Cases > Vessels & Trucks

- Increase interoperability between physical assets
- Digitize ownership of physical assets
- Enable remote control of autonomous assets

Digital Twins

Concept

Embedded systems offer "digital tethers" in the physical realm

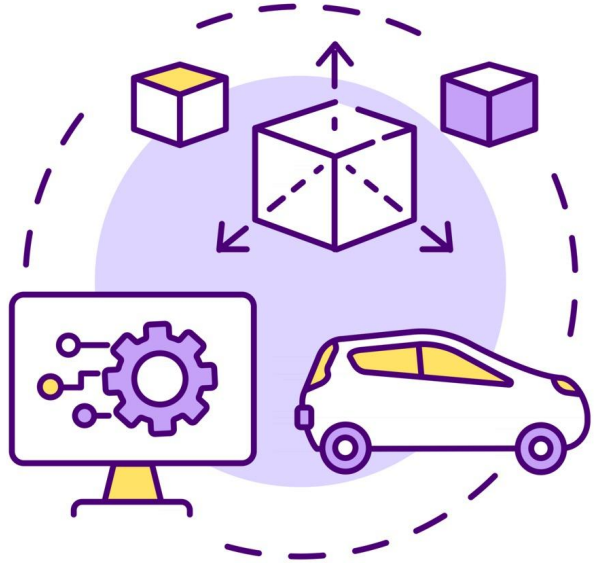


Businesses operate digital twins to manage physical assets

Digital twins proxies "digital tethers" in the virtual realm

Digital Twin Prototype

- Design prototype for digital twins
- Used to explore assets in virtual reality
- Not linked to existing physical assets
- Explicit states improve modularity



Digital Twin Prototype

Digital Twin Instance

- Linked to a unique physical asset
- Associated with a single prototype
- Relies heavily on eventual consistency
- Processes huge amounts of data

Digital Twin Aggregate

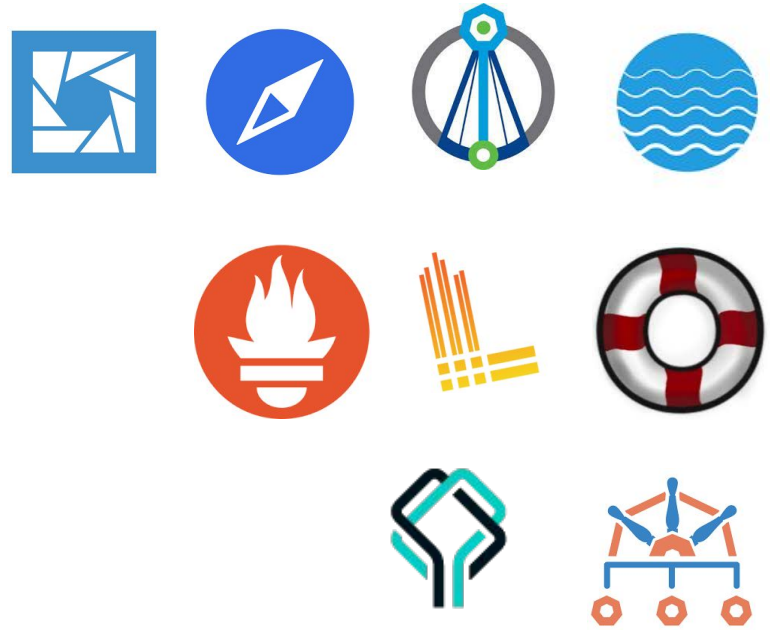
- Aggregate of other digital twins
- Construct has access to all its dependencies
- Interleaves dependencies to improve heterogeneity
- Delegates “Command & Control” operations

DIGITAL TWIN AGGREGATE

Digital Twins with Crossplane

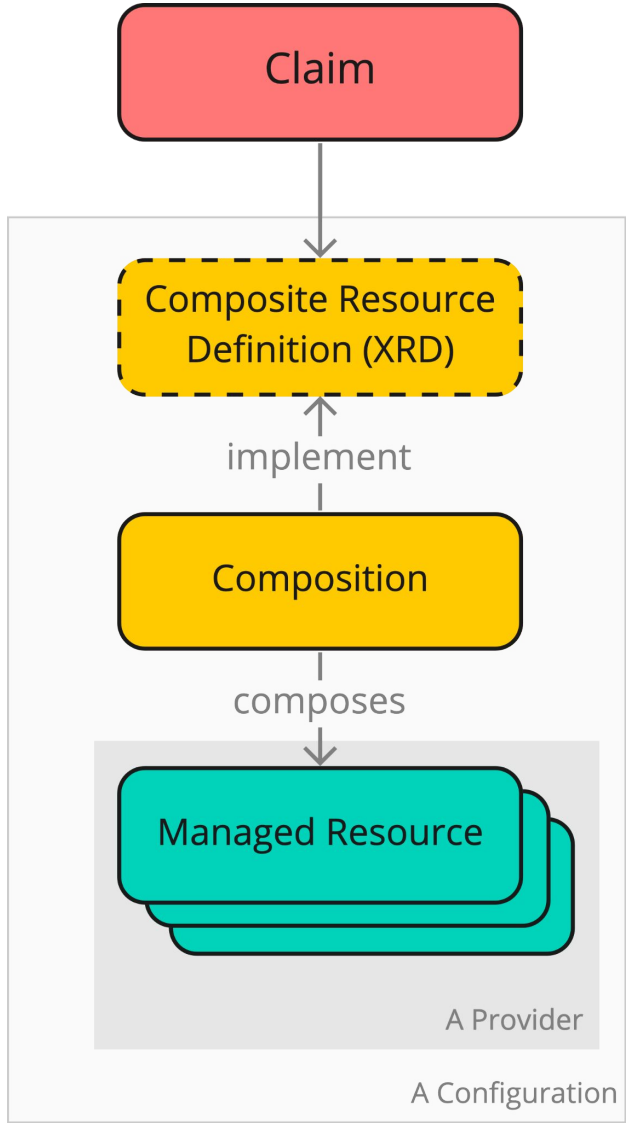
K8s as an IoT Platform

- Declarative Model, KRM, Controller
- Runtime available everywhere
- Rich Ecosystem
  - Clients for Digital Twins operators: Lens, Octant, kubenav
  - Backup & Restore: Velero
  - Observability: Prometheus, Loki, Robusta
  - Policies: Datree, Kyerno

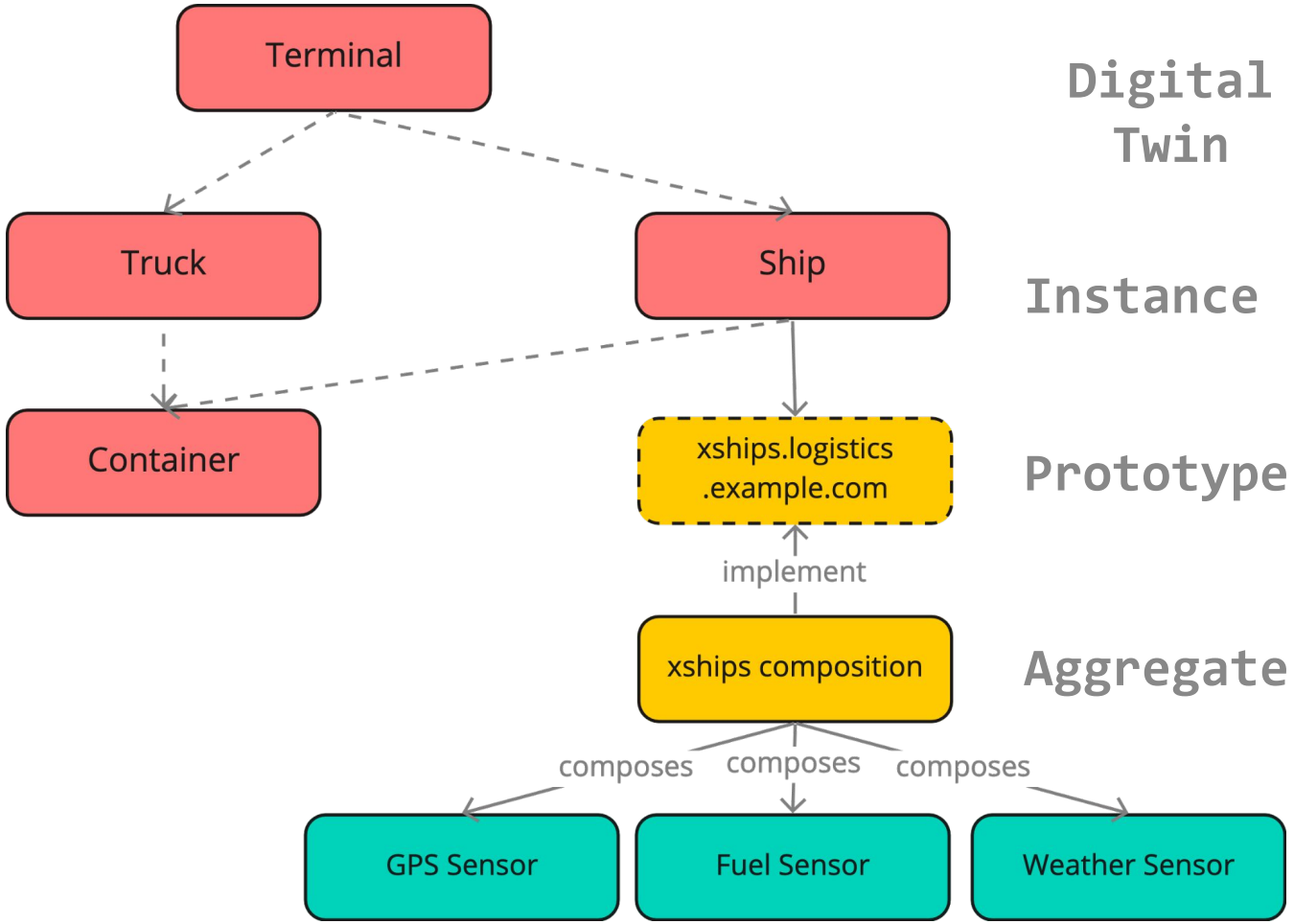


Crossplane

- Crossplane and the Kubernetes Ecosystem provides us with an incredible platform that we can use to build and operate digital twins.
- MR: Make Cloud resources available in K8s
  E.g. local sensors, cloud services
- XR: Compose multiple cloud resources
  Compose different Sensors into different digital twin representations



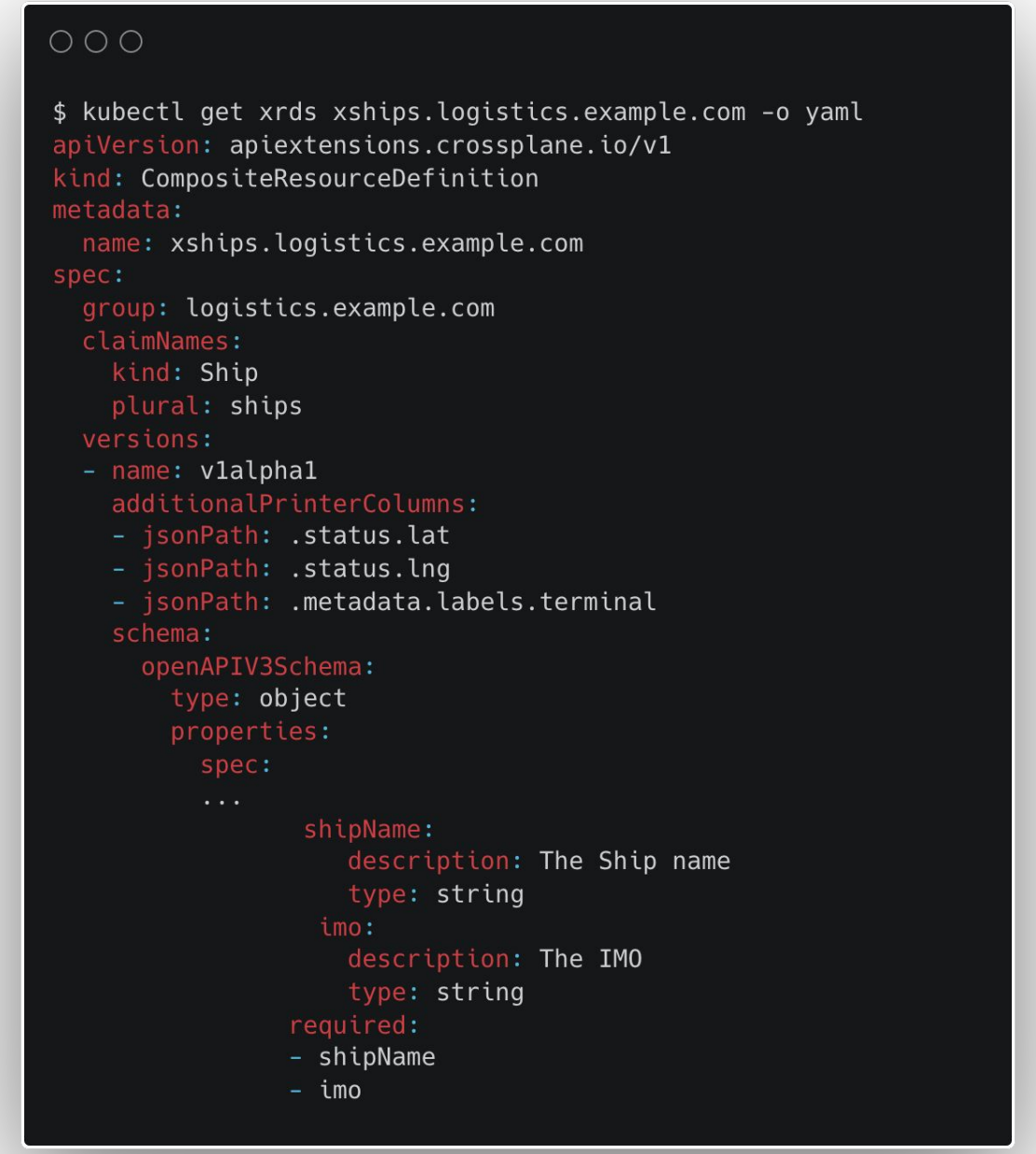
A Logistics Platform



Source github.com/luebken/platform-example-logistics

Demo

API Definition



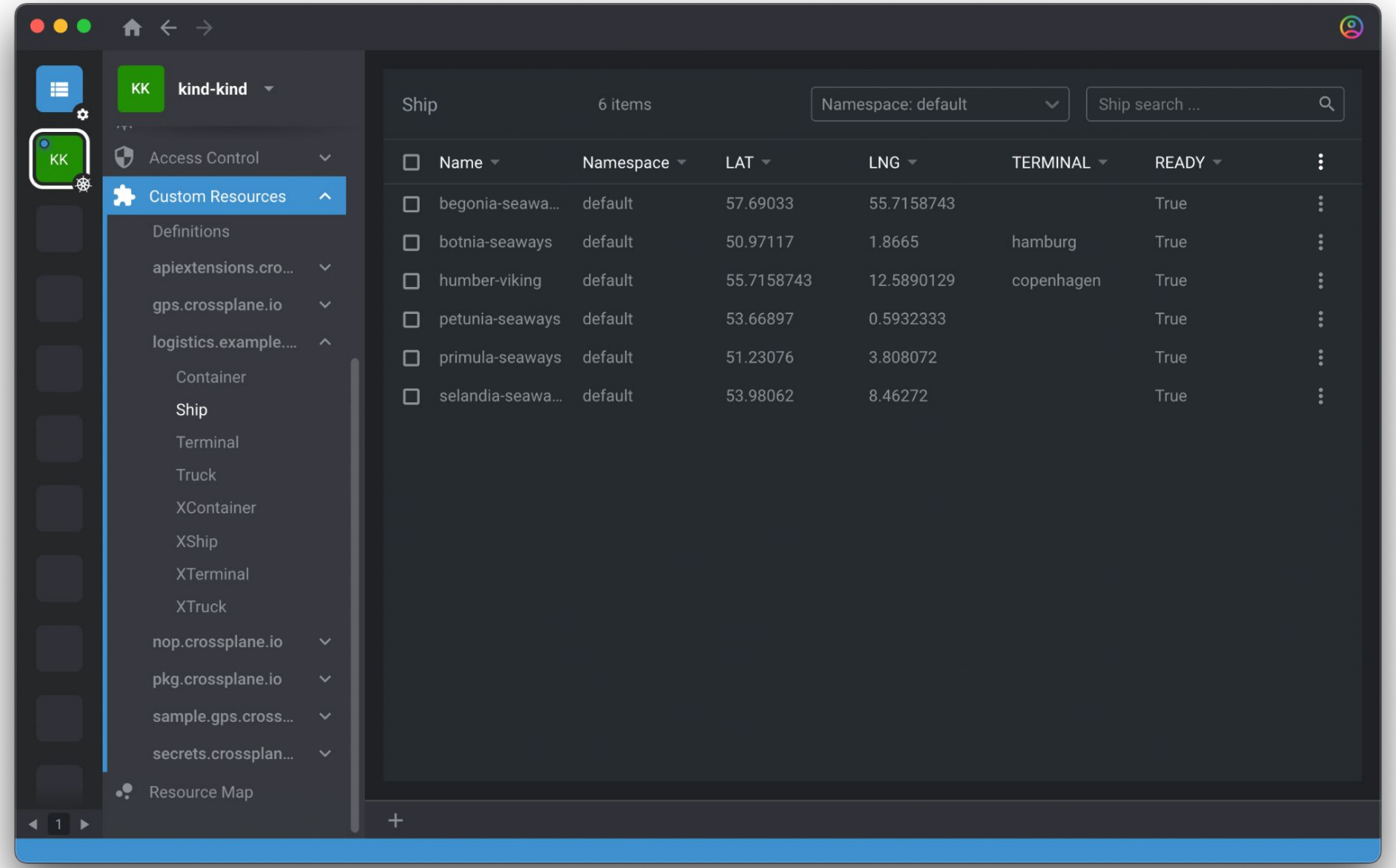
```
$ kubectl get xrds xships.logistics.example.com -o yaml
apiVersion: apiextensions.crossplane.io/v1
kind: CompositeResourceDefinition
metadata:
  name: xships.logistics.example.com
spec:
  group: logistics.example.com
  claimNames:
    kind: Ship
    plural: ships
  versions:
  - name: v1alpha1
    additionalPrinterColumns:
    - jsonPath: .status.lat
    - jsonPath: .status.lng
    - jsonPath: .metadata.labels.terminal
    schema:
      openAPIV3Schema:
        type: object
        properties:
          spec:
          ...
                 shipName:
                    description: The Ship name
                    type: string
                  imo:
                    description: The IMO
                    type: string
                required:
                - shipName
                - imo
```

Query with kubectl

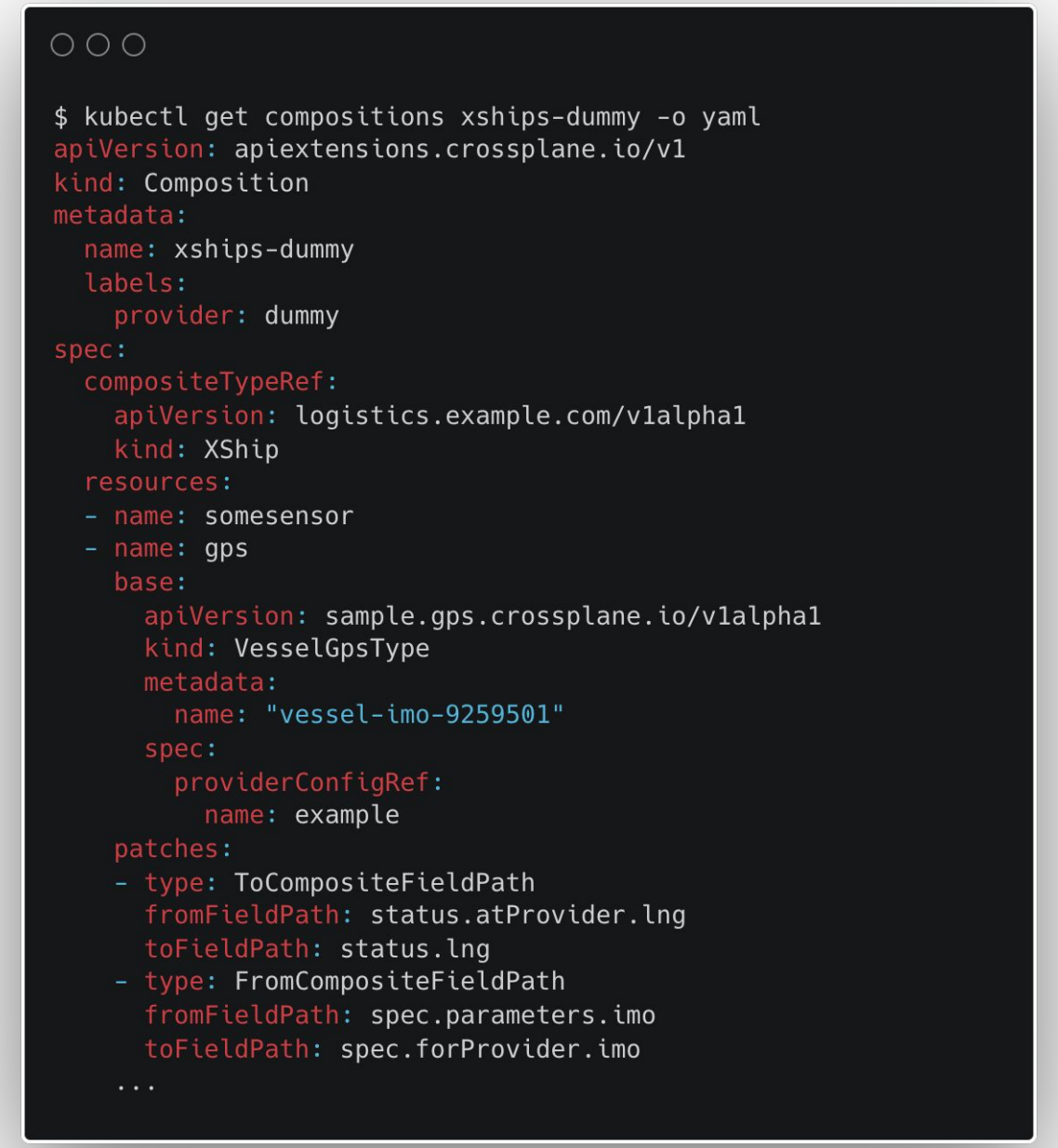

```
$ kubectl get ships,terminals
NAME                                      LAT          LNG          TERMINAL     READY   AGE
ship.logistics.example.com/begonia-seaways    57.69033     55.7158743                True    17s
ship.logistics.example.com/botnia-seaways     50.97117     1.8665       hamburg      True    17s
ship.logistics.example.com/humber-viking      55.7158743   12.5890129   copenhagen   True    17s
ship.logistics.example.com/petunia-seaways    53.66897     0.5932333                 True    17s
ship.logistics.example.com/primula-seaways    51.23076     3.808072                  True    17s
ship.logistics.example.com/selandia-seaways   53.98062     8.46272                   True    17s

NAME                                      COUNTRY   READY   AGE
terminal.logistics.example.com/brevik         Norway    True    17s
terminal.logistics.example.com/brugge         Belgium   True    17s
terminal.logistics.example.com/copenhagen     Denmark   True    17s
terminal.logistics.example.com/fredrikstad    Sweden    True    17s
terminal.logistics.example.com/hamburg        Germany   True    17s
terminal.logistics.example.com/immingham      UK        True    17s
```

Query with Lens



Composition



```
$ kubectl get compositions xships-dummy -o yaml
apiVersion: apiextensions.crossplane.io/v1
kind: Composition
metadata:
  name: xships-dummy
  labels:
    provider: dummy
spec:
  compositeTypeRef:
    apiVersion: logistics.example.com/v1alpha1
    kind: XShip
  resources:
  - name: somesensor
  - name: gps
    base:
      apiVersion: sample.gps.crossplane.io/v1alpha1
      kind: VesselGpsType
      metadata:
        name: "vessel-imo-9259501"
      spec:
        providerConfigRef:
          name: example
    patches:
    - type: ToCompositeFieldPath
      fromFieldPath: status.atProvider.lng
      toFieldPath: status.lng
    - type: FromCompositeFieldPath
      fromFieldPath: spec.parameters.imo
      toFieldPath: spec.forProvider.imo
    ...
```

Managed Resources


```
$ kubectl get managed
NAME                                                                  READY   SYNCED   LAT          LNG
vesselgpstype.sample.gps.crossplane.io/begonia-seaways-6h7mn-mf74f    True    True     57.69033     55.7158743
vesselgpstype.sample.gps.crossplane.io/botnia-seaways-bm284-64qjm     True    True     50.97117     1.8665
vesselgpstype.sample.gps.crossplane.io/humber-viking-5wd57-jppnn      True    True     55.7158743   12.5890129
vesselgpstype.sample.gps.crossplane.io/petunia-seaways-zcp26-xkqk4    True    True     53.66897     0.5932333
vesselgpstype.sample.gps.crossplane.io/primula-seaways-vdgtg-6zp56    True    True     51.23076     3.808072
vesselgpstype.sample.gps.crossplane.io/selandia-seaways-28lqw-58tb4   True    True     53.98062     8.46272
```

Provider

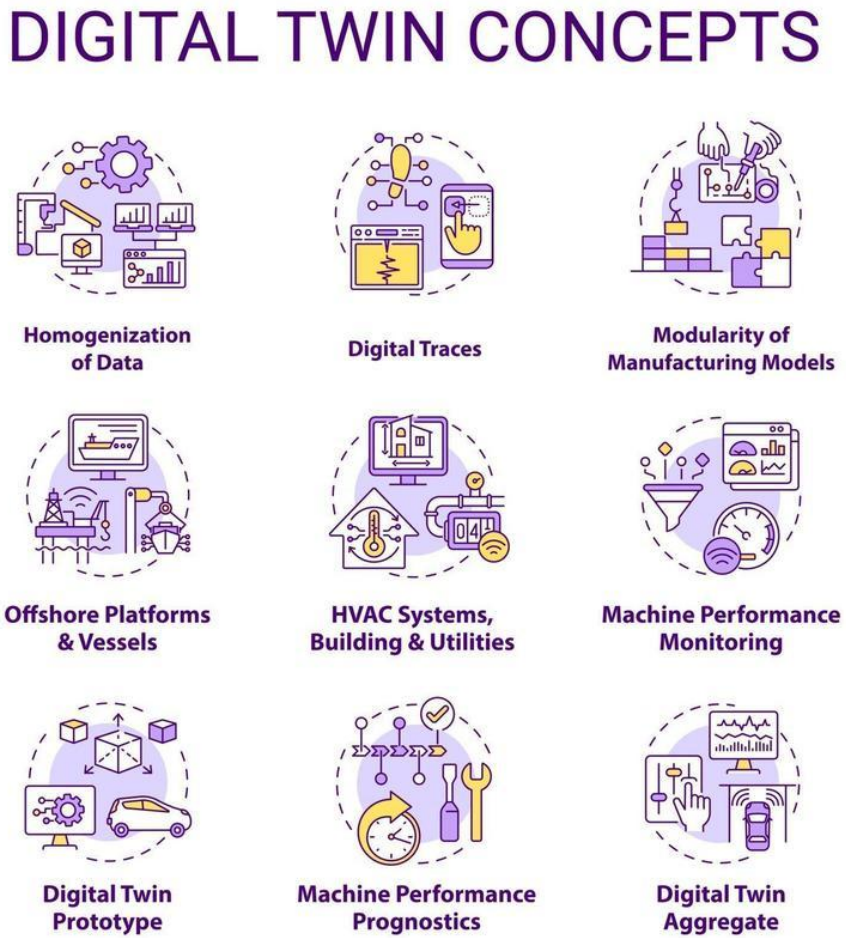
```
$ kubectl get crds vesselgpstypes.sample.gps.crossplane.io -o yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: vesselgpstypes.sample.gps.crossplane.io
spec:
  group: sample.gps.crossplane.io
  names:
    kind: VesselGpsType
    listKind: VesselGpsTypeList
    plural: vesselgpstypes
    singular: vesselgpstype
  scope: Cluster
  versions:
  - additionalPrinterColumns:
    - jsonPath: .status.conditions[?(@.type=='Ready')].status
    - jsonPath: .status.conditions[?(@.type=='Synced')].status
    - jsonPath: .status.atProvider.lat
    - jsonPath: .status.atProvider.lng
    name: v1alpha1
    schema:
      openAPIV3Schema:
        description: A VesselGpsType is an example API type.
        properties:
          apiVersion: ...
          kind: ...
          metadata: ...
          spec:
            properties:
              forProvider:
                properties:
                  imo:
                    type: string
                required:
                - imo
```

Source github.com/luebken/provider-gps-dummy/

Conclusion

Conclusion

- Digital twins must overcome challenges & limitations in technologies and people to succeed
- Digital twins benefits greatly from the abstraction layer the Kubernetes platform provides
- Digital twins can easily provision & manage complex cloud infrastructure via Crossplane
- Digital twins are ideal for containerization
- Next Steps:
  - Finalize Integration with IoT HuB for Sensor Simulation @ https://github.com/dfds/digital-twin-poc



KubeCon CloudNativeCon Europe 2022

Q&A

Find us in Pavilion 2
Crossplane booth - CNCF Area
Upbound booth - Orange Zone S104

Tobias Andersen, DFDS

Matthias Lübken, Upbound
@luebken / @upbound_io