From the Cloud to the Clouds

Taking Integrated Modular Avionics on the Next Level with Cloud-Native Technologies

by Christian Rebischke

Agenda

- $ whoami
- The Problem
- The Research Question
- The Forecast
- The Discussion

$ whoami

- Linux User since 12 years
- Arch Linux Contributor since 8 years
- Google Summer of Code Alumni (Secure Software Supply Chains)
- Github: https://github.com/shibumi
- Blog: https://shibumi.dev
- Former Work Student in the TUC Datacenter, ISSE, Institute for Mathematics
- Currently employed as Site Reliability Engineer (while finishing my thesis)

The Problem

- Planes and Data centers have a few things in common:
  - **Efficiency**:
    - cost efficiency (high profit, high costs for equipment)
    - power efficiency (less energy consume, good workload saturation, climate protection..)
    - place efficiency (planes need room for payload, less cables, etc ..)
    - process efficiency (software should succeed)
  - **Reliability**: website customers want uptime, passengers do not want to die
  - **Security**: a hacked website can cost billions of $, a hacked plane could be worse
  - **Costs:** both are expensive -> Urge to shrink costs
  - **THEY ARE BOTH FLYING!** yes not kidding (linux computers on the Mars (rovers), in Satellites, in spaceships, drones, planes)
- So what is the difference? -> slow certification processes, aviation runs behind ‘state-of-the-art’ technologies on the ground, etc.

The Problem

Figure 2 - IMA Cabinet Components

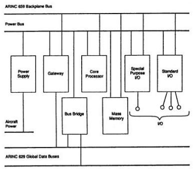
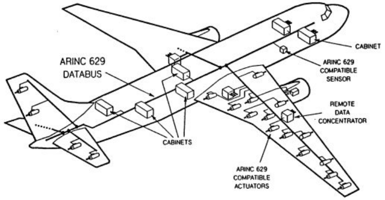


Figure 1 - Typical Aircraft Configuration



- The Distributed Flight Software Architecture “Centuries”
  - The Age of “Federated Systems” (before ~2000):
    - Many LRU/LRM (Line-Replaceable Units/Modules) -> very expensive
    - many cables, distributed/”wasted” CPU/RAM power -> wasted capacity/weight
    - Software especially written for that LRU or Data Concentrator etc -> very static
  - The Age of “Integrated Modular Avionics” (after ~2000):
    - Contracting CPU/RAM-Power into Cabinets (**ARINC 653**)
    - Communication via Data/Global bus (**ARINC 629**)
    - Standards to packaging and maintenance (**ARINC 650/651**)
    - -> Similarities to Service Oriented Architecture, modern Microservice orchestration,..
- How will the Future look like? It’s not written yet, thus we can shape it!

The Problem = Bureaucracy?!

- Strict Certification Processes are mitigating change
- => Certifications as major blocker for bleeding edge technologies
- We do not want to look for holes in the certification process
- More research in that area establishes confidence
- Possible certifications to look at:
    - DO-178C
    - DO-254/ EUROCAE ED-80
    - DAL A/B/C/D/E (these are mostly hardware related?! What about Software requirements?!)
    - Similarities between Aviation Certification and Linux on Automotives certification?!
    - ARINC standards
    - STANAG standards (NATO)
    - ASAAC standards (Def Stan 00-74 for IMA for example)

Research Question(s)

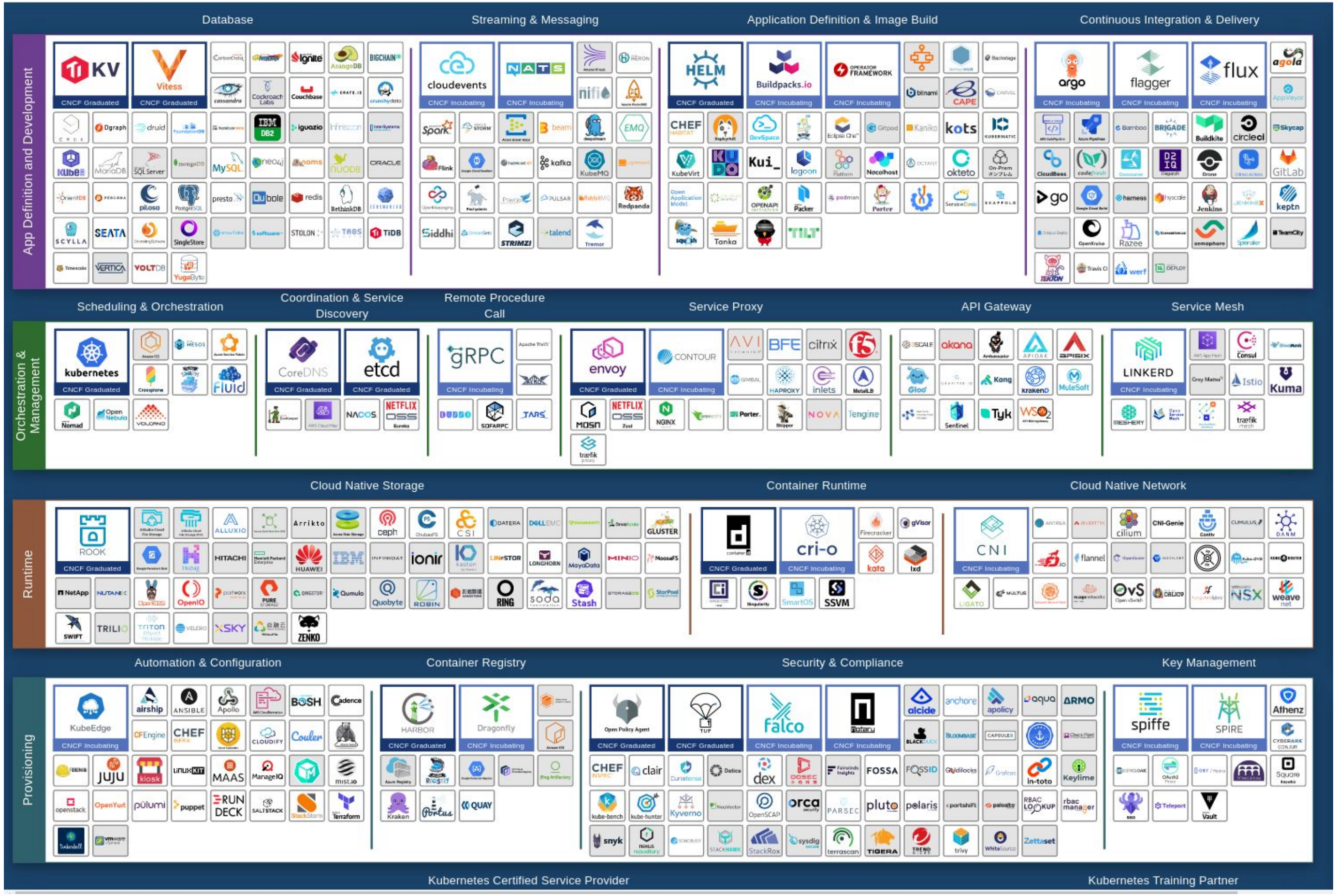
- How can the next age look like?
- Can we create synergies between fields? Share knowledge? Be more cost effective? Can we re-use existing knowledge/software/applications?
- What would we gain with small certification changes (Real-Time Linux, more open source, …)?
- How can we make flight and/or cabin software more reliable and secure?

I want to try to answer them all

Forecast

- Due to my Container/Linux background I would like to focus on comparing aviation software architecture with cloud-native architecture.
- The US military experiments with Cloud-Native technologies already: Kubernetes (a container orchestration) in Planes (U-2 spy plane, F16, etc)
- Problem: The research about this topic is top secret and there seem to be no record about civil research on this area -> It’s worth it.
- There is plenty of research to do …..

The Cloud-Native Landscape: Many possibilities for Aviation for software delivery, messaging, orchestration, networking, runtime ...


App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Kubernetes Certified Service Provider
Kubernetes Training Partner

Forecast

- Many similarities: Container/Virtual Machines <-> Separation Kernels, ARINC 653 Partitioning etc
- Use of Linux as Real-Time OS (SpaceX is already using it!) opens a huge, new ecosystem -> DO-178C finally needs to address this certification. We can give hints and work to make this possible for DO-178D(?!)
- New concepts for the orchestration of Aviation software:
  - Lessons learned: What did we learn on the ground, what could be useful for the clouds?
  - Service Meshes, Continuous Integration/ Continuous Delivery for Flight Software Development, Security Concepts, performance/saturation enhancements, etc..

References

- Pictures on Slide 5 from: Prisaznuk, Paul J. "Integrated modular avionics." *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference@ m_NAECON 1992*. IEEE, 1992
- Picture on Slide 8 from: https://landscape.cncf.io (visited 2029-04-21)
- For all references see: https://github.com/shibumi/master-thesis/blob/main/thesis/literature.bib

The Discussion - Questions/Suggestions/Changes/etc