



**LYNX**

SOFTWARE  
TECHNOLOGIES



**THOMPSON SOFTWARE  
SOLUTIONS**

A LYNX SOFTWARE TECHNOLOGIES COMPANY



**timesys**

A LYNX SOFTWARE TECHNOLOGIES COMPANY

# REALIZING VISION OF TRULY REUSABLE SOFTWARE IN MISSION CRITICAL SYSTEMS

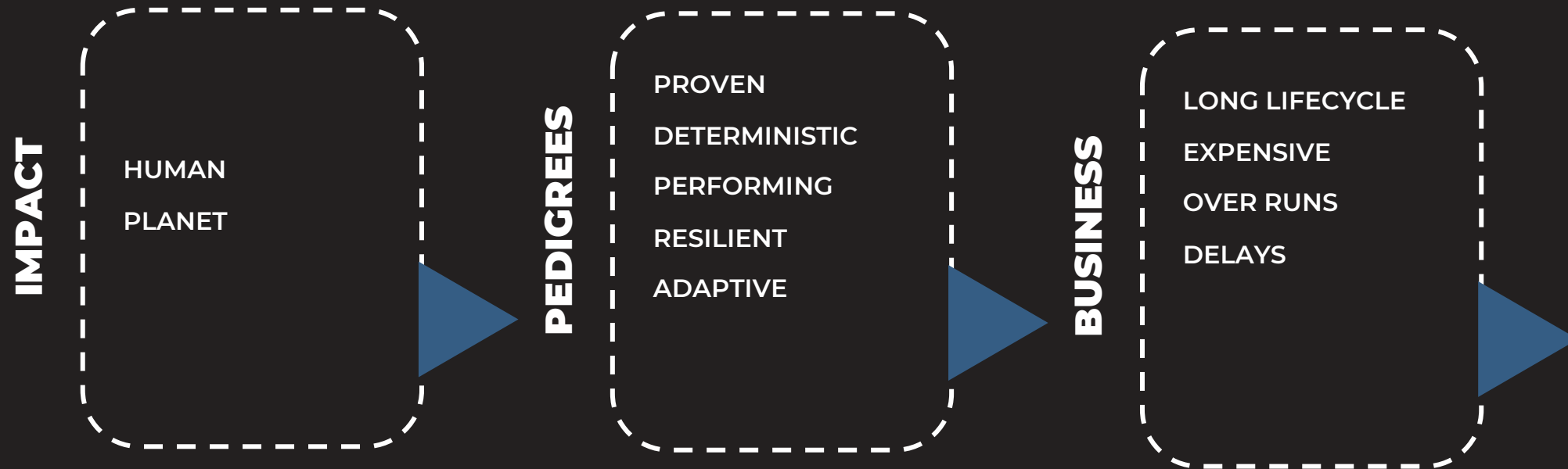
---

Will Keegan, CTO | Lynx

Michel Genard, Board member & Strategy | Lynx

# SETTING THE CONTEXT OF MISSION CRITICAL SYSTEM

WHERE GOOD ENOUGH IS NOT GOOD ENOUGH!



# MARKET PERSPECTIVE

## SOFTWARE VENDOR

- Open
- Seamless
- All-in-one

## OEM

- Differentiator
- Manage cost

## PRIME

- Enough to win the program
- Expectations Management

## GOVERNMENT

- Vendor Lock!
- Not fast enough
- Overruns

# OPEN | REUSE | PORTABILITY

NOT REALLY NEWS, RIGHT?

**VME**

Modular Open Systems Approach

**MOSA**

**SOSA**<sup>TM</sup>  
Sensor Open Systems Architecture

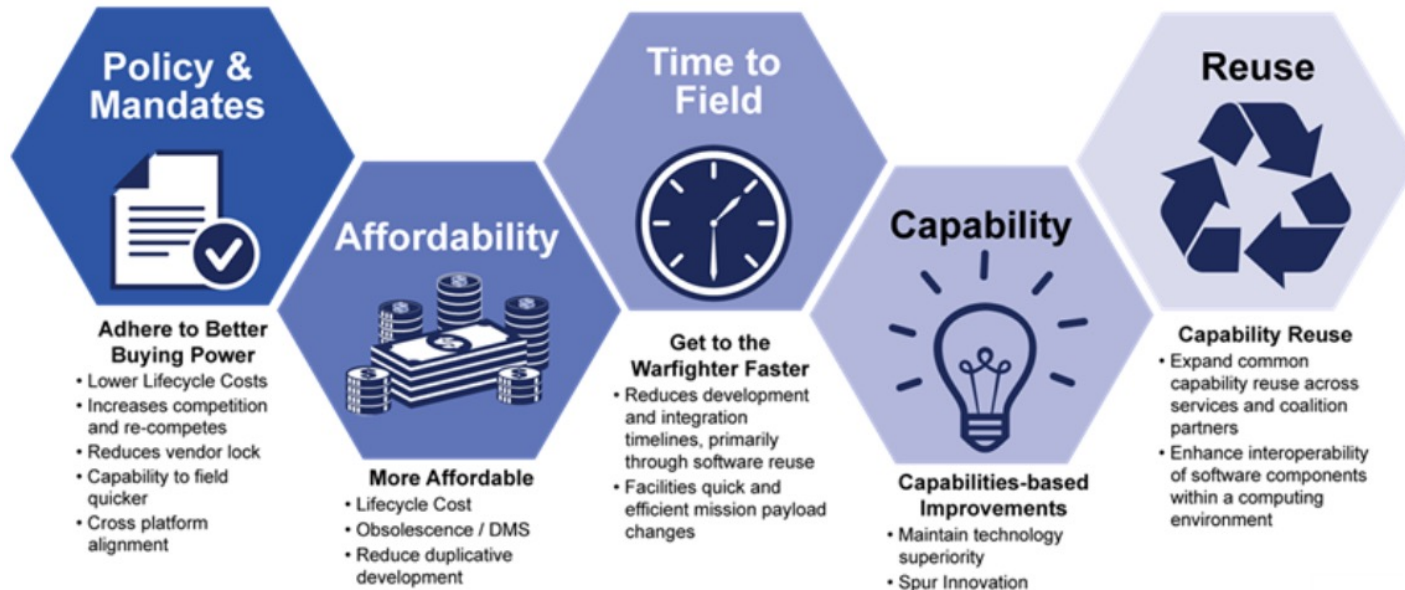
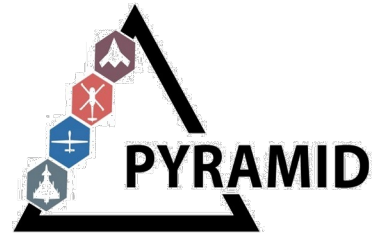
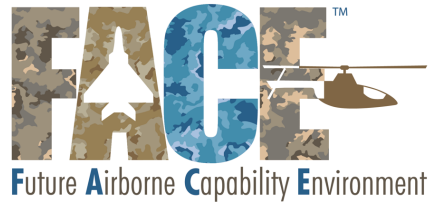
**PCI**



**virtio**

# NOW ENTER MOSA

## STEM FROM PLATFORM COMPOSSIBILITY



<b>Adoption:</b>	<ul style="list-style-type: none"> <li>Widely adopted software architecture of choice</li> <li>International OSA alignment</li> <li>Open &amp; competitive marketplace</li> <li>Adaptable &amp; rapid response to threats</li> <li>Reduced costs and time to deployment</li> </ul>
<b>Governance:</b>	<ul style="list-style-type: none"> <li>PYRAMID is owned by the UK MOD</li> <li>Governed by a management agency led by UK MOD</li> <li>Enables shared benefits &amp; efficient certification</li> <li>Controlled access to a digital support environment</li> <li>Component proprietary to remain with developer</li> </ul>
<b>Marketing &amp; Promotion:</b>	<ul style="list-style-type: none"> <li>Promote as best practice on a global scale</li> <li>Work with selected international partners</li> <li>Evidence based demonstrations &amp; successes</li> </ul>
<b>Skills:</b>	<ul style="list-style-type: none"> <li>Sustainable pool of skilled engineers</li> <li>Affordable training pipeline</li> <li>Flexible engineering workforce</li> </ul>

## MOSA objectives

PROGRAM EXECUTIVE OFFICE  
AVIATION

- Reduce supply chain risk**  
Competitive options, obsolescence mitigations, and simplified logistics tail.
- Reduced schedule pressure**  
Quicker and more flexible contracting options and reduced repetitive NRE efforts.
- Increased affordability**  
Avoid vendor lock, increase competition, and enable reuse (develop & certify once, deploy many).
- Enhanced capabilities**  
Lower barrier to entry for technology insertions by strategically creating "windows" of capability insertion with defined key interfaces.
- Increased readiness**  
Increased speed to field, simplified logistics tail, increases cyber resiliency.

```
systemresources>
  <processor process
</systemresources>
<schedulingpolicy policy
  <majorframe process
    <dynamicssched>
      <thread v
        <thread v
      </dynamicssched
    <minorframe vp
  </majorframe>
</schedulingpolicy>
<partition pname="partit
<guestimage memregionnam
path="/home/lsb/lynxsecu
<guestimage memregionnam
path="/home/lsb/lynxsecu
<guestimage memregionnam
<guestimage memregionnam
<guestimage memregionnam
server lab/busylsa/lsa64
<guestimage memregionnam
path="/home/lsb/lynxsecu
<externaldevice external
externaldevname="CLK8" p
externaldevname="CLK10"
externaldevname="FIRMWAR
externaldevname="RESET0"
externaldevname="SERIAL1
vdevname="VIRTGICO" dev
vdevname="VIRTGICO" dev
```

# DISCONNECTS VISION VS REALITY



## OPERATING ENVIRONMENT

Platform portability

---



## DEVELOPMENT PARADIGM

Heterogenous development

---



## MOSA USE CASE

Re-use

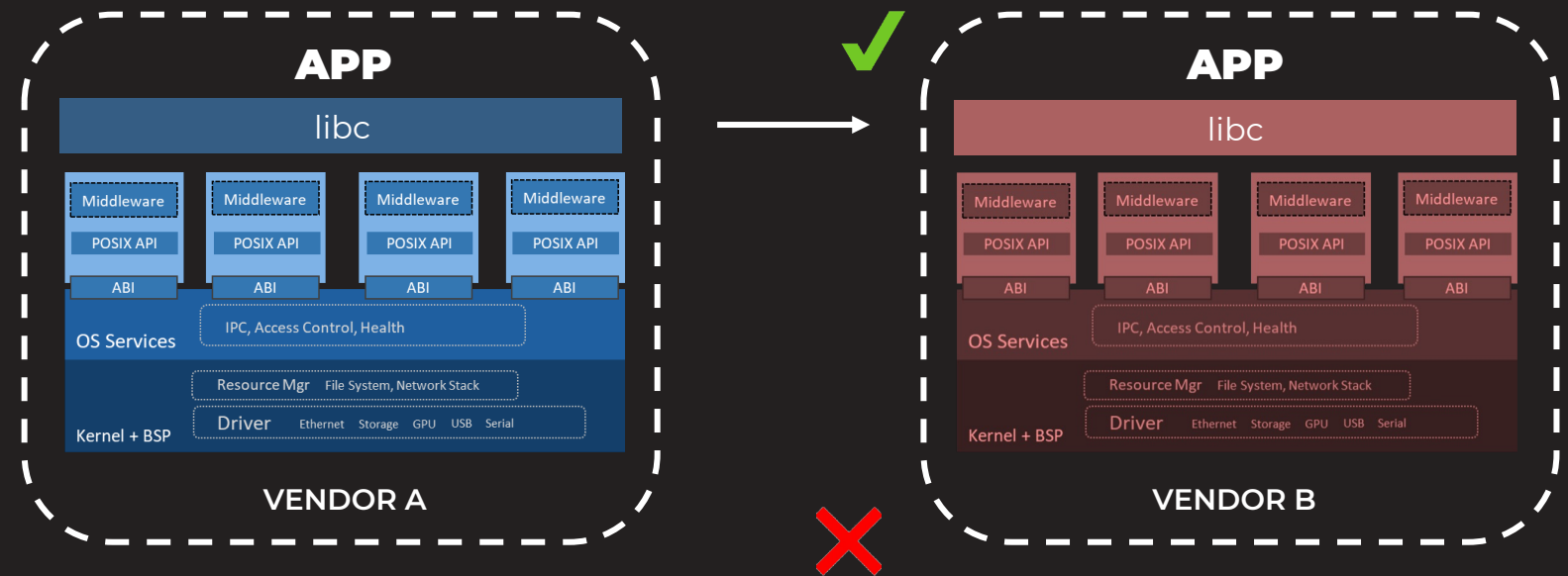
# PLATFORM PORTABILITY FACTORS

## APP LEVEL

- Platform API “libc”

## PLATFORM LEVEL

- Unique Platform Behavior
- Unique Driver Architecture
- Proprietary
  - BSP
  - System Configuration
- System Integration Tooling
- Analysis/Evaluation Artifacts



Incompatible

- Kernel libraries
- Subsystem coupling

**THE  
FUTURE IS  
BRIGHT!**



**LYNX**  
SOFTWARE TECHNOLOGIES

# MOSA

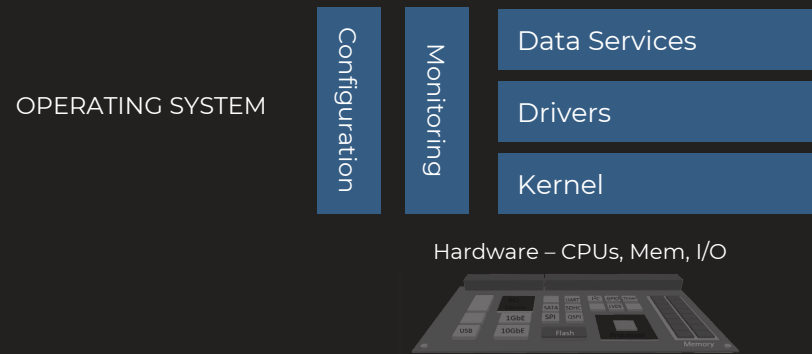
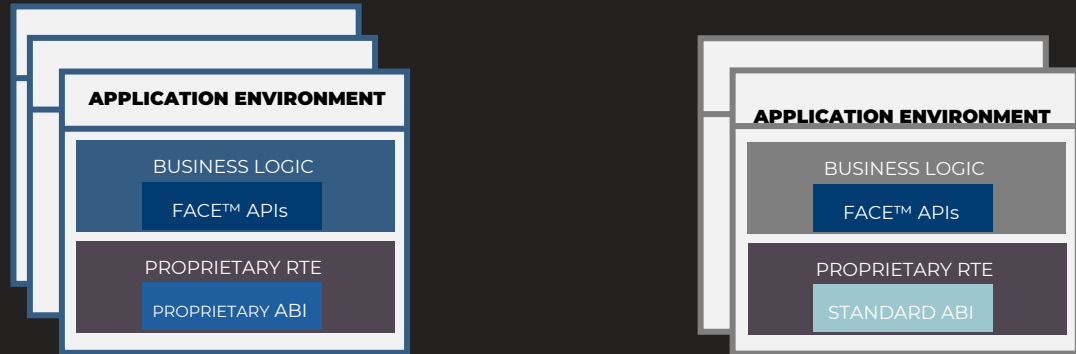
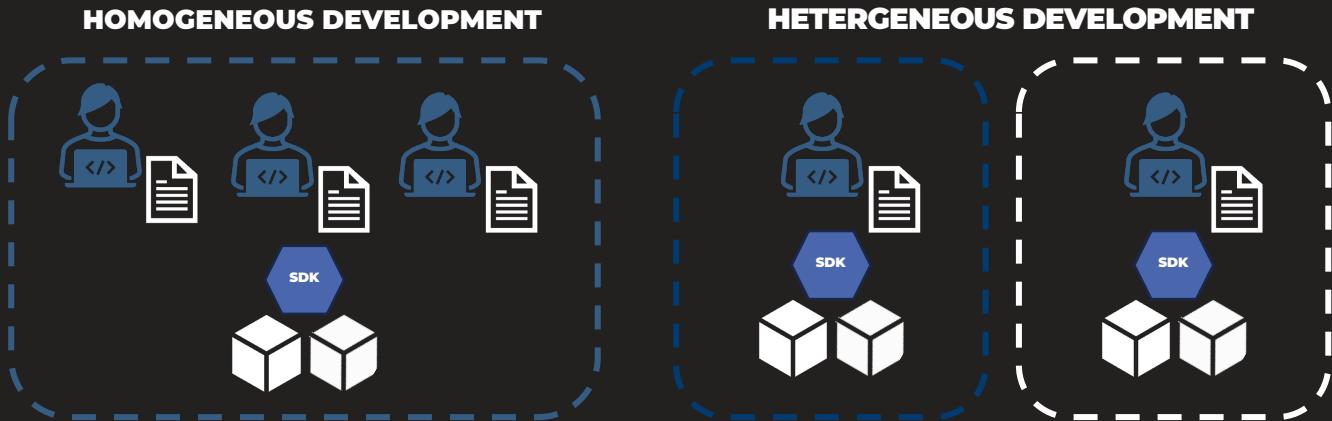
## FOR OPERATING SYSTEMS

### TODAY

- Homogenous SDK

### FUTURE

- Heterogeneous SDK
- Separable OE Layers
- Standard Configuration

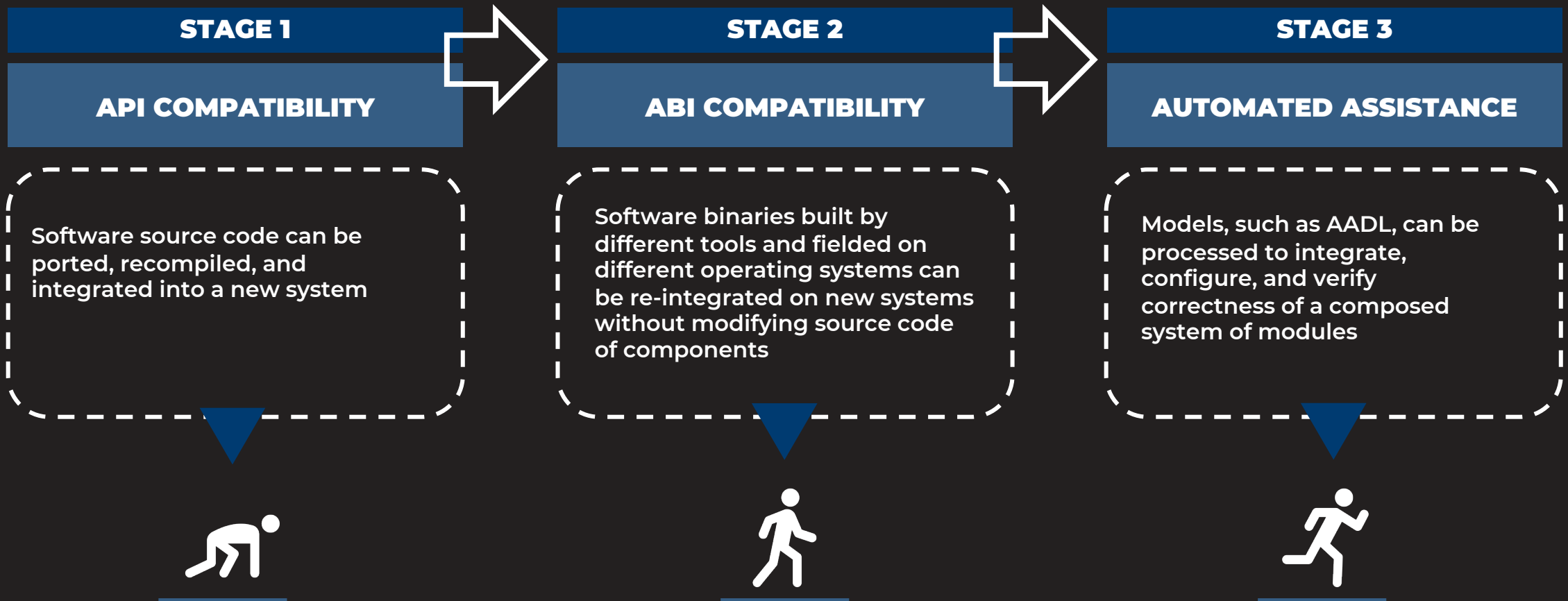


# COMPOSITION REDINESS LEVEL CONCEPT

Composition readiness level (CRL) is a systematic measurement of the interfacing of compatible interactions between components on the same Operating Environment. The methodology can help discover a direction for improving integration with other software components.

Composition Readiness Levels	
Levels	Description
9	Composed systems can be validated with automation
8	Composition of system can be automated
7	Configuration of component can be automated via standard specs
6	Components be configured with parameters via standard interfaces
5	Components can interoperate with different generation components and OEs
4	Components can run at any level of a software stack
3	Components can be independently tested on surrogate lab environments
2	Components can have different attributes (language, addressing)
1	Systems Integrated with components from different SDKs
0	Components can be built from a common set of APIs

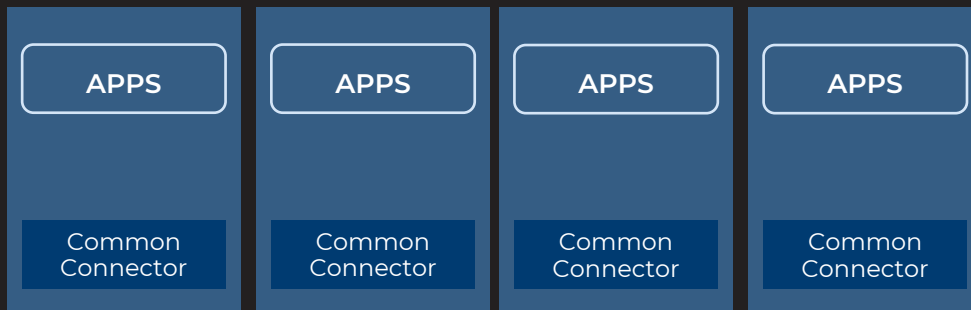
# MARKET PERSPECTIVE



# PORTABILITY

## SK+UNIKERNEL ARCHITECTURE

### UNIKERNELS

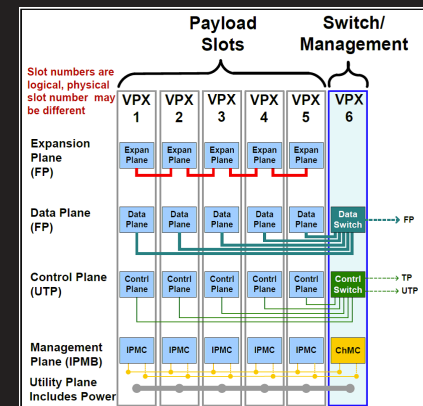


Standard Application Module Backplane

Separation Kernel / Hypervisor

### GOALS

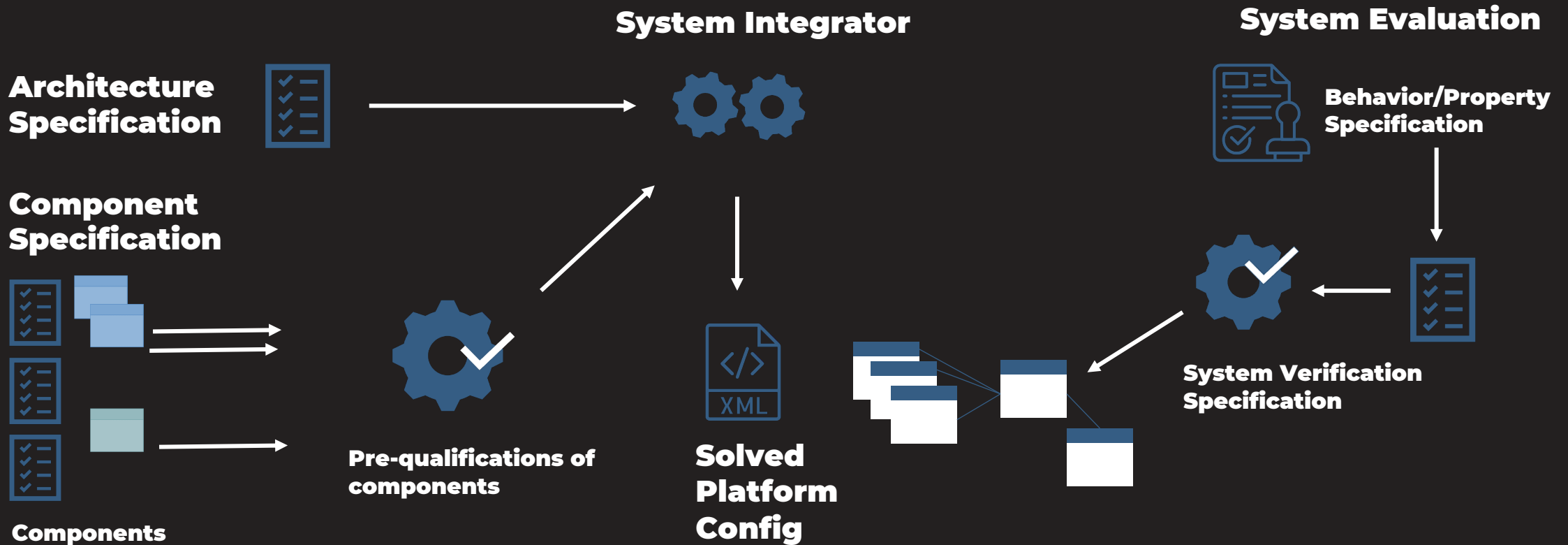
- Preserve current API dependency
- Enhance Portability
  - Run on Linux
  - Run on Qemu
  - Run on Hypervisor
  - Run on Cloud Platform / Containers
- Open up applications
  - Let customers have source access to all application dependencies
- Make separation kernel a standard software backplane  
*Like a VPX Backplane*



- Standard device tree interface
- Standard Data Plane
- Standard Plug-n-Play
- Standard Mgmt
- Standard Diagnostics

# MBSE ASSISTANCE

MODEL GUIDED INTEGRATION AND EVALUATION OF PREVIOUSLY DEVELOPED COMPONENTS



```
<systemresources>
  <processor processo
</systemresources>
<schedulingpolicy policy
  <majorframe process
    <dynamic Sched>
      <thread v
        <thread v
      </dynamic Sched
    <minorframe vp
    <minorframe -
  </majorframe>
</schedulingpolicy>
<partition pname="partit
<guestimage memregionnam
path="/home/lsb/lynxsecu
<guestimage memregionnam
path="/home/lsb/lynxsecu
<guestimage memregionnam
<guestimage memregionnam
<guestimage memregionnam
server lab/busylsa/lsa64
<guestimage memregionnam
path="/home/lsb/lynxsecu
<externaldevice external
externaldevname="CLK8" p
externaldevname="CLK10"
externaldevname="FIRMWAR
externaldevname="RESET0"
externaldevname="SERIAL1
vdevname="VIRTGICO" dev
vdevname="VIRTGICO" d
```

# TAKEAWAYS

## VISION OF TRULY REUSABLE SOFTWARE



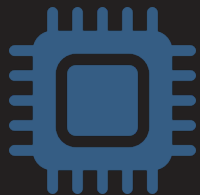
### SYSTEM LEVEL PERSPECTIVE

- Heterogenous Development
- From virtual to real hardware



### TRUTH SOFTWARE DISAGGREGATION

- Clear design rules
- Small units of execution



### SHIFT LEFT

- Heterogenous Development
- From virtual to real hardware

# BRINGING THE VALUE OF OPEN STANDARDS

TO AVIONICS & DEFENSE

REUSE OF EXISTING  
CODE



AVOIDS VENDOR  
LOCK-IN



ACCELERATE SYSTEM  
DEPLOYMENT TO  
WARFIGHTERS



**SOLUTIONS THAT NOT ONLY MEET TODAY'S CHALLENGES BUT PAVE THE WAY FOR  
TOMORROW'S TRIUMPHS.**