# CYBER ASSURED SYSTEMS ENGINEERING

SEL4 SUMMIT 11 OCTOBER 2022

DARREN COFER



UNITED STATES ARMI

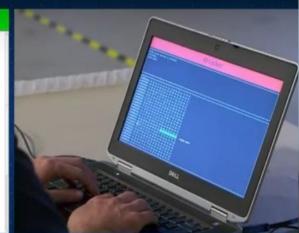
CASE



# DARPA HACMS

#### HIGH ASSURANCE CYBER MILITARY SYSTEMS





Loonwerks.com/projects/hacms April 2017







▶ ● 2:19 / 3:43



< pm

🦛 ::

CC

GSOOH





We brought a hackable quadcopter with defenses built on our HACMS program to @defcon #AerospaceVillage. As program manager @raymondrichards reports, many attempts to breakthrough were made but none were successful. Formal methods FTW!



10:20 AM · Aug 9, 2021 · Hootsuite Inc.



### CYBER ASSURED SYSTEMS ENGINEERING (CASE)

Develop model-based systems engineering tools and workflow to make the HACMS approach repeatable, scalable, more incremental

#### Design-in cyber-resiliency

- Automated architecture transforms for threat mitigation
- High assurance components generated from specifications
- Techniques to deal with legacy code ("cyber retrofit" using virtual machines)

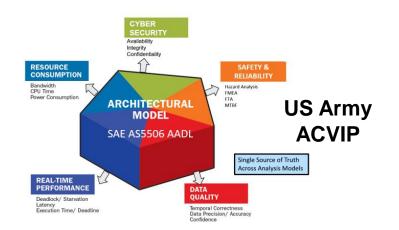
#### Build what you model

- Build system directly from detailed, verified AADL model
- Make the security guarantees of seL4 accessible to system developers
- Ability to target different platforms to facilitate incremental debugging/development

#### Provide evidence

- Formal verification of functional and cyber-resiliency properties, information flow properties, component proofs
- Code generation equivalence to model, seL4 build preserves properties
- Integrate evidence as an assurance case demonstrating how/why requirements are satisfied







### BRIEFCASE INTEGRATED WORKFLOW

#### WITH INTEGRATED ASSURANCE

llins Aerospace

- 1. Capture/import cyber-resiliency requirements based on initial AADL model analysis (GearCASE and DCRYPPS)
- 2. Transform system architecture model to satisfy cyberresiliency requirements
- 3. Generate new high-assurance components from formal specifications (SPLAT) or pre-verified libraries
- 4. Verify system design using **formal methods** (AGREE) and information flow analysis (Awas)
- 5. Checks model conformance to standards (Resolint)
- 6. Generate **software integration code** (HAMR) directly from verified architecture models, targeting multiple operating systems (including seL4)

AM

TEST

RADIO

UAV

FILT

FPLN

AM

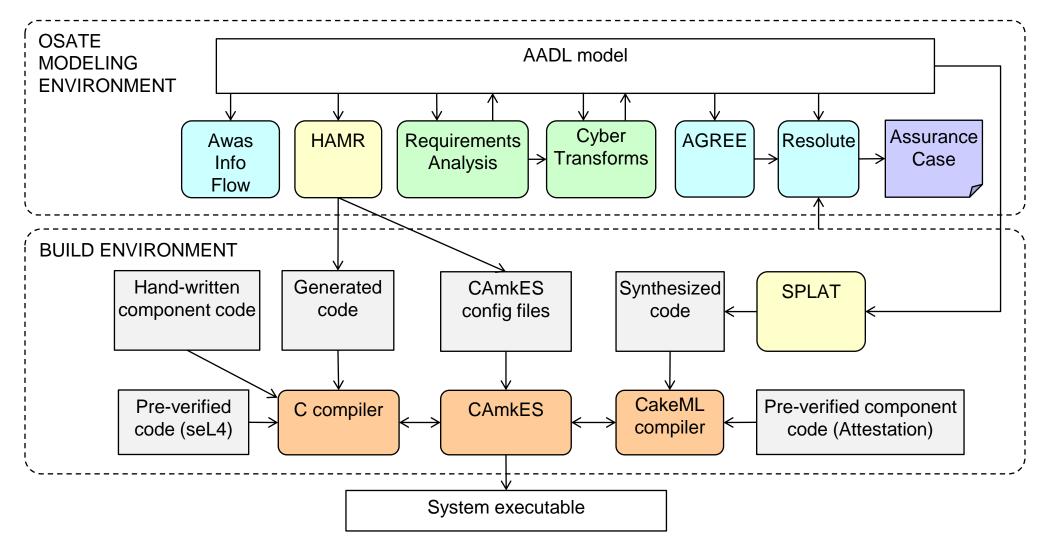
↓ID\_List

7. Document evidence/compliance with assurance case (Resolute)

<pre>with CASE_Props;  with CA</pre>	<pre>with CASE_Props; system Nup failures maintenance: in event data port; map:out event data port; map:out event data port; map:request: in event data port; filight_plan: out event data port; filight_plan: out event data port; map: in event data port; map: data event data port; map: data event data port; map: data event data port; map: data event deta port; map: data event deta port; map: data event data port; map: data event event data port; map:</pre>	pac ⊝pub	ckage Aircraft blic	with AADL
<pre> system Map features maintenance: in event data port; map:request: in event data port; mop:request: in event data port; forogeneties CASE_Props::Trust_Level =&gt; Untrusted; end Pijs system NappointNanager features grist in event data port; mop:request: out event data port; m</pre>	system Map signed reader data port; map:request: in event data port; map:request: in event data port; properties CASE_Props::Trust_Level >> Untrusted; end HightPlan features (CASE_Props::Trust_Level => Trusted; end HightPlans request: out event data port; map:request: out event data p	pub		An Introduction to the SAE
<pre>features features fightplan features fightplan; out event data port; properties GSE_Props::Trust_Level =&gt; Untrusted; end Nap; System FlightPlan features fightplan; out event data port; properties GSE_props::Trust_Level =&gt; Trusted; end FlightPlan; System Kaysinffanager features fightsplan: in event data port; mp: in event data port; fightplan: in event data port; mp: in event data port; fightplan: in eve</pre>	<pre>features maintenance: in event data port; map: equest: in event data port; properties CASE Props::Trust_Level =&gt; Untrusted; end FlightPlan features pilot_input: in event data port; map: equest: out event data port; map: in event data port; m</pre>			Architecture Analysis & Design Language
<pre>maintenance: in event data port; map.request: in event data port; properties CASE_Props::Trust_Level =&gt; Untrusted; end Pilpt input: in event data port; properties CASE_Props::Trust_Level =&gt; Trusted; end FilghtPlans features gris in event data port; map.request: out event data port; features featu</pre>	<pre>maintenance: in event data port; msp: out event data port; msp: caguest: in event data port; properties CASE_Props::Trust_Level =&gt; Untrusted; end PlightPlan features cASE_Props::Trust_Level =&gt; Trusted; end FlightPlan; system RaypointManager features msp: in event data port; msp: in event da</pre>			
<pre>msp: out event data port; properties CASE_Props::Trust_Level =&gt; Untrusted; end Hap; system FlightPlan features pilot_input: in event data port; properties CASE_props::Trust_Level =&gt; Trusted; end FlightPlan; features gps: in event data port; msp: request: out event data port; msp: request: out event data port; flight_plan: in event data port; msp: request: out event data port; ffight_plan: in event data port; ffight_plan: in event data port; msp: freetures featu</pre>	<pre>mp: creduets: in event data port; properties CASE_Props::Trust_Level =&gt; Untrusted; end Map; system FlightPlan features filight_plan: in event data port; map: in event</pre>			
<pre>map_request: in event data port;</pre>	<pre>map_request: in event data port; CASE_Props::Trust_Level =&gt; Untrusted; end Nap; system FlightPlan features CASE_Props::Trust_Level =&gt; Trusted; end FlightPlan; system KaypointManager features: map_request: in event data port; map_request: in event data port; map.request: unt event data port; map.reque</pre>			
<pre>properties cdSE_Props::Trust_Level =&gt; Untrusted; end Map; system FlightPlans features pilot_input: in event data port; properties cdSE_Props::Trust_Level =&gt; Trusted; end FlightPlans features gp: in event data port; map:request: out event data port; feat features f</pre>	<pre>properties GASE_Props::Trust_Level =&gt; Untrusted; end Nap; system FlightPlan features pilot_input: in event data port; properties GASE_Props::Trust_Level =&gt; Trusted; end FlightPlan; system kaypointNanager features gp::in event data port; map:request: out event data port; ma</pre>			
<pre>CASE_props::Trust_Level &gt;&gt; Untrusted; end Map; system FlightPlan features</pre>	CASE props::Trust_Level => Untrusted; end Nap; system FlightPlan features more read CASE props::Trust_Level => Trusted; end FlightPlan; system NaypointNanager features more request: out event data port; map_request: out event data port; Tight_plan: in event data port; Tight_plan: in event data port; the test avert data port; map_request: out event data port; the test avert data port; the test avert data port; feat system F feat feat feat feat feat feat feat feat			
<pre>system flightPlan features flight_plan: out event data port; mop.createsticatesti</pre>	system FlightPlan features d&E Props::Trust_level => Trusted; end FlightPlan; gps: in event data port; mpp: nequest: out event data port; mpp: nequest: out event data port; mpp: nequest: out event data port; msp: nequest: out event data port; fight_plan: in event data port; msp: nequest: out event data port; fight_plan: in event data port; msp: nequest: out event			
<pre>features     properties     cose properti</pre>	<pre>features properties CASE Proper:Trust_Level =&gt; Trusted; end FlightPlan; system WaypointHanager features gps: in event data port; map: request; out event data port; map: request; out event data port; map: nevent data port;</pre>		end Map;	
<pre>features     properties     cose properti</pre>	<pre>features properties CASE Proper:Trust_Level =&gt; Trusted; end FlightPlan; system WaypointHanager features gps: in event data port; map: request; out event data port; map: request; out event data port; map: nevent data port;</pre>			
pilet input: in event data port; properties CASE_Props::Trust_Level => Trusted; end FightPlan; properties gps: in event data port; mp:request: out event event event data port; mp:request: out event event	<pre>plot_input: in event data port; properties CASE_Props::Trust_level =&gt; Trusted; end FiliphtPlans; system NaypointNanager; features map.request: out event data port; map.request: out event data port;</pre>			
<pre>filight_plan: out event data port; properties GASE_Props::Trust_Level =&gt; Trusted; end FlightPlan; system WaypointNanager fister     gps: in event data port;     map:nequest: out event data port;     map:nequest: out event data port;     map.in event</pre>	<pre>figit_plan: out event data port; properties GASE Props::Trust_Level =&gt; Trusted; end FlightPlan; system KaypointNanger features gps: in event data port; map_request: out event data port; suppointNanger; end WaypointNanger; system F feat Fight_plan: in event data port; system F feat Fight_plan:</pre>			
<pre>properties CASE_Props::Trust_Level =&gt; Trusted; end FlightPlan; esystem WaypointHanager features gp: in event data port; map:request: out event data port; waypoint: out event data port; map:request: out event data port; map:request: out event data port; map:request: out event data port; map:request: out event event data port; map:request: out event e</pre>	CASE Propost: Trust_Level => Trusted; end FlightPlan; system WaypointManager features gps: in event data port; map: in event data port; map: in event data port; itaght.plan: in event data port; filight.plan: in event data port; features system F features			
CASE_Props::Trust_Level => Trusted; end FlightPlan; system WaypointManager features gps: in event data port; map: in event data port; map: in event data port; flight_plan: in event data port; flight_plan: in event data port; flight_plan: in event data port; end WaypointManager; system F feat end Fligt end Fligt end Fligt Fat Fat Fat Fat Fat Fat Fat Fa	CASE Props::Trust_Level => Trusted; end FlightPlan; system WaypointNanager features gps: in event data port; map:request: out event data port; tightplan: in event data port; flightplan: event data port; flig			
<ul> <li>system NaypointManager features</li> <li>gps: in event data port; map: nequest: out event data port; map: nequest: out event data port; filight_plan: in event data port;]</li> <li>end WaypointManager;</li> <li>System F</li> <li>System F</li> <li>System F</li> <li>System I</li> <li>System</li></ul>	system WaypointWanager features gpr: in event data port; mp: request: out event data port; mypointWanager; flight_plan: in event data port; system Ker feat end WaypointWanager; system I feat in the new Cogen hasts lack flags field of the Cose SmithNak Widev its feat in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its feat in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cogen hasts lack flags field of the Cose SmithNak Widev its in the new Cose SmithNak Widev i			Peter H. Feiler
<ul> <li>system Kaypoint/Hanager features</li> <li>gps: in event data port; map_request: out event data port; map_request: out event data port; map_request: out event data port; map_request out event data port; map_request</li></ul>	system WaypointNanager features gps: in event data port; map:equest: out event data port; waypoint: out event data port; Tight_Dan: in event data port; system F feat end WaypointNanager; system I feat in the way Dagane (Mar-accelerated information of the Dises Endshare Widey Has feat in the way Dagane (Mar-accelerated information of the Dises Endshare Widey Has feat in the way Dagane (Mar-accelerated information of the Dises Endshare Widey Has feat in the way Dagane (Mar-accelerated information of the Dises Endshare Widey Has feat in the way Dagane (Mar-accelerated information of the Dises Endshare Widey Has end Flig system I in the ways of the Dises Carl a bit of the Dises Endshare (Mar-accelerated information of the Dises Endshare (Mar-a		end FlightPlan;	
features gps: in event data port; map: in event data port; map: in event data port; flight_plan: in event data port; map: in event data port; flight_plan: in event data port; flight_cont Waysimit_plan: i	features gps: in event data port; mp:request: out event data port; mp/fight_plan: in event data port; msypointHanger; system F feat end KappointHanger; end Fligi system T feat			David P. Gluch
<pre>system I feat</pre>	gps: in event data port; map_request: out event da			
System I Alert Alert Alert FC Alert Al	Bill event data port; map_request: out event data port; maypoint: out event data port; Tight_plan: in event data port; rend KaypointManager; system F feat end Flig System I System I Feat			
Alert	Alert			
Flight_plan: in event data port;     I       end WaypointManager;     Image: I	flight_plan: in event data port;       I         end WaypointManager;       end WaypointManager;         system F       The take way Oragen hasgits zerb Trave Red Analysis CARE for 0 Case Sensitivities Under Heg         feat       File         end Fligg       File         system I       File         Station       File			
end WaypointNanager; Constant & Sector State - State	end WaypointManager; System F, Feat Feat Feat Feat Feat Alert FC FC FC		contractions and accord data accord.	
System F     The Life way 20 again hauges back haused up high block     Feet	Alert			
<ul> <li>system F, fie ble ver "Degen hargets beet Payer Polet Antiges CARE for D-Cass Sensitivary Weder Her</li> <li>feat</li> <li>Field</li> <li>Fi</li></ul>	Alert		flight_plan: in event data port;	I
Alert	Alert		flight_plan: in event data port; end WaypointManager; system F feat	I Anges GATE for 8-Sec SendsHars Weday Hes Course of Process of the Course of the Res Q of a
Alert	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I form forgraps: GART: for 0-Case Sample Mara Window Heg 21 Control (1997) (2007)
Alert FC FC → FC	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I from CATC for 0-Case Sample Mana Window Hea Count Area (Count Action 10 Count Action 10 Co
Alert FC FC Alert	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I have begins CATE for 0-Care Sends New Weday Files Care and P and Care Sends New Weday Files Care and P
Alert	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I from Margas CSATE flue 0-Case Sanata Mara Window Hea 2007 A 400 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Alert FC FC	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I Inter Interpret CATE for 0-Case Sensis Mara Weday Hea Case A Cate Cate Cate Cate Cate A
Alert FC Alert A	Alert		flight_plan: in event data port; end Waypoint/Manager; System F feat feat end Flig end Flig	I Inter Inter CARL for 0-Cas SandsMara Wedey Hes Cas A Cas D Control Cas SandsMara Wedey Hes Cas A Cas D Cas Sands Mara Wedey Hes Cas D Cas A Cas D Cas Sands Mara Wedey Hes Cas D Cas
Alert	Alert		flight_plan: in event data port; end WaypointManager; System F; feat; H Lds ww Cogner he gis keth Faywa Pokt Feat; H H H H H H H H H H H H H H H H H H H	I Internet CARE for 0-Case Sensis Mara Wedev Hea Party of 0 + E E 2005 + columno Case Access P 0 P 0 P 0 P 0 P 0 P 0 P 0 P 0
Alert	Alert		flight_plan: in event data port; end WaypointManager; system F feat end Flig system I	I Inter
Alert	Alert		flight_plan: in event data port; end WaypointManager; Converte Northering Converte Nor	I Inter
Alert Alert FC	Alert		flight_plan: in event data port; end WaypointManager; System F; feat feat feat fight_plan: in event data port; feat feat feat fight_plan: in event data port; feat feat feat feat feat feat feat feat	I free tenjes CSATE for 0-Case SanateMara Wedge Hes Case Access Constant Access Constant Access Constant Case Access P & Case Constant Access Constant Case Access P & Case Constant Access P & Case Acce
Alert	Alert		flight_plan: in event data port; end WaypointManager; System F; feat feat fight_plan: in event data port; feat fight_plan: in event data port; fight_plan: in event data port; fight_plan: eve	I Internet CEATE for 0-Case SensiteMara Window Heis Particle A Case SensiteMara Window Heis Particle A Case SensiteMara Window Heis Particle A Case SensiteMara Window Heis Case A Case A Case Sensite L Case Sensite
Alert	Alert		flight_plan: in event data port; end WaypointManager; system F, feat feat system Ti end Flig system I www.seators.com fight_plan: fight feat feat feat feat feat feat feat fea	I Inter
Alert	Alert		flight_plan: in event data port; end WaypointManager; Convert Aventmeries (CAR) - 3 - charactering files/file feat feat feat file in a convert for the set file in a convert for the set f	I Inter
FC Free Prov	FC FC Vide Störmporarti		flight_plan: in event data port; end WaypointManager; System F; feat feat feat fight_plan: in event data port; feat feat fight_plan: in event data port; fight_plan: in the fight fight_plan: magnetic fight_plan: fight_plan: fight_plan: fight_plan: magnetic fight_plan: fight_plan: fight_plan: fight_plan: magnetic fight_plan: fight_plan: fight_plan: fight_plan: fight_plan:	I from Verges CSATE flor 0-Case SensiteMara Window Heis Photos - Case Sensite Photos
FC	FC Voix	0	flight_plan: in event data port; end WaypointManager; System F, The tak way Cogner ha gis Back Faywar Podd Feat Feat Flight Start Start Control (1997) Film F Start Contro	I Internet CEATE for 0-Case SensiteMara Wedev Here Particles A Care Constant Accord Wedev Here Care Accord Particles Constant Accord Particles Care Accord Particles Care Accord Noordal Nordal N
C Mark	⊡ Vesik ⊡ Subornportati	0	flight_plan: in event data port; end WaypointManager; System F, The tak way Cogner ha gis Back Faywar Podd Feat Feat Flight Start Start Control (1997) Film F Start Contro	I tere tergen CARE for 0.5 sends Mars Weder Hes Control of the 0.5 sends Ma
C Mark	⊡ Vesik ⊡ Subornportati	0	flight_plan: in event data port; end WaypointManager; System F, The tak way Cogner ha gis Back Faywar Podd Feat Feat Flight Start Start Control (1997) Film F Start Contro	I tere tergen CARE for 0.5 sends Mars Weder Hes Control of the 0.5 sends Ma
	E Subserperaris	0	flight_plan: in event data port; end WaypointManager; System F; feat feat feat fight system I; end Flig system I; fight fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Av	I tere tergen CARE for 0.5 sends Mars Weder Hes Control of the 0.5 sends Ma
/	E Subserperaris		flight_plan: in event data port; end WaypointManager; System F; feat feat feat fight system I; end Flig system I; fight fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Av	I Inter
	4 E Weeksteel	~	flight_plan: in event data port; end WaypointManager; System F; feat feat feat fight system I; end Flig system I; fight fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent aventure 27 b fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Flig varpoint fight_plan: Avent Avent Avent Avent Avent Flig varpoint fight_plan: Avent Av	I form tengen CSATE The 0-Case SensiteMara Wedger Hes Case Accord PC Control Contro

Model-Based

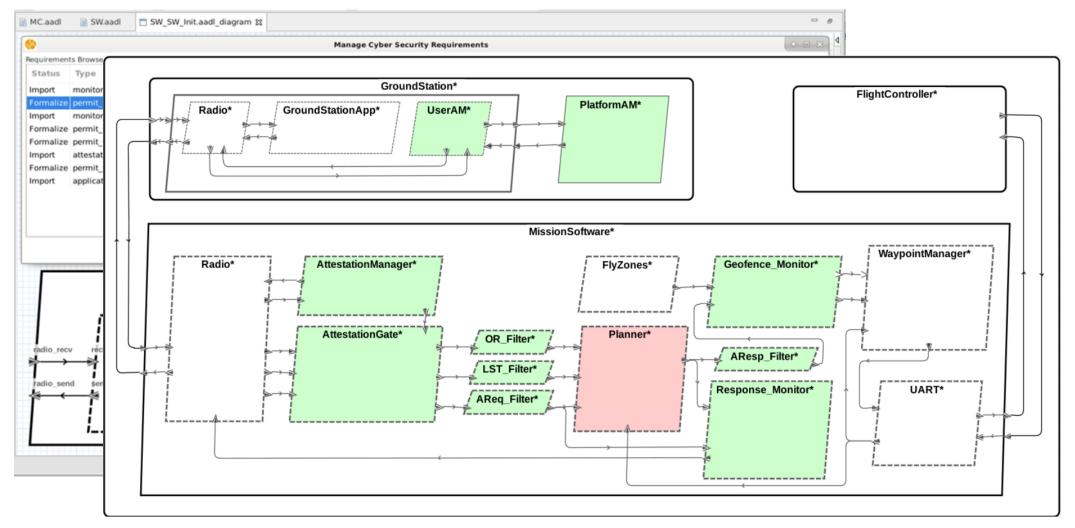
### BRIEFCASE TOOL ARCHITECTURE





### SYSTEM ARCHITECTURE TRANSFORMATION

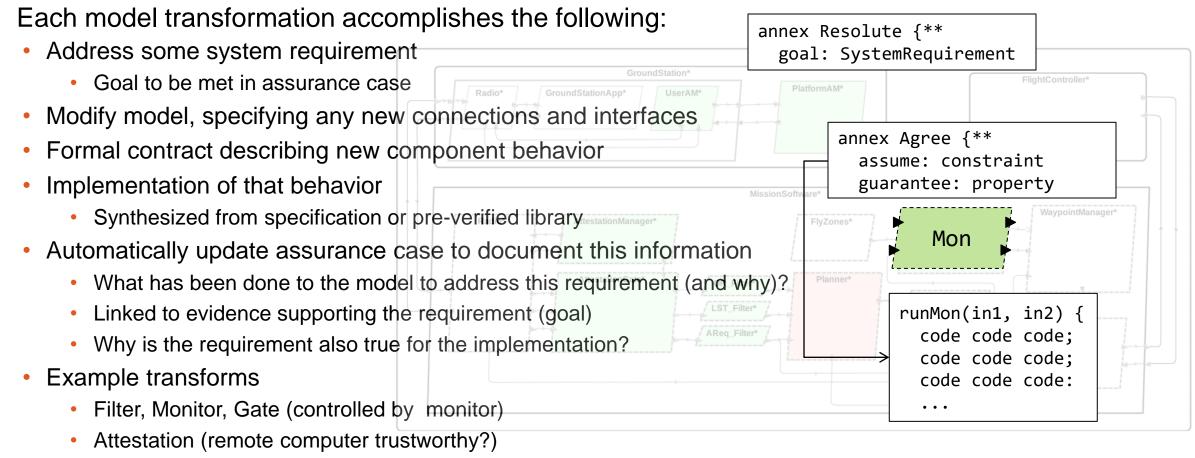
#### EXPERIMENTAL PLATFORM : SMALL UAV





### SYSTEM ARCHITECTURE TRANSFORMS

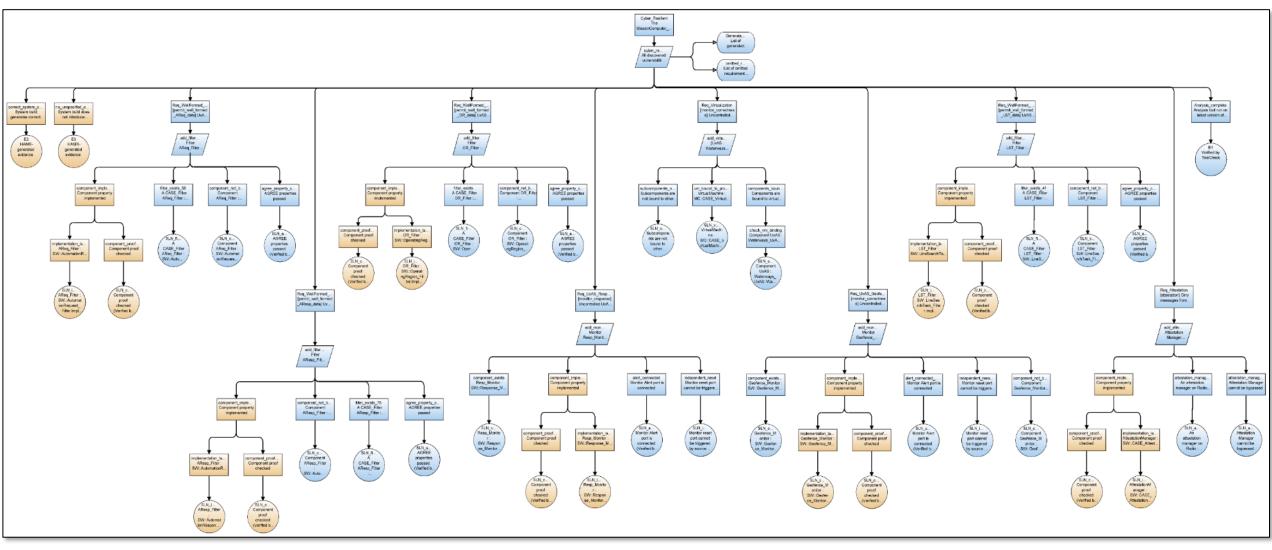
#### ASSURANCE CASE BUILT AUTOMATICALLY



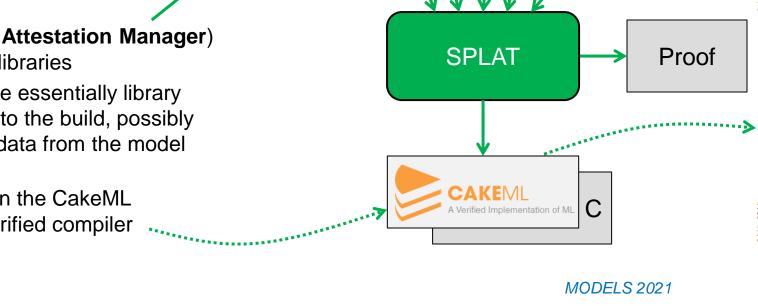
• Virtualization, seL4 build prep



## RESOLUTE ASSURANCE CASE





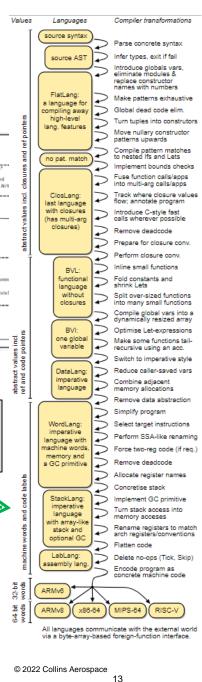


Line Search

FLT L ST

# GENERATE HIGH ASSURANCE COMPONENTS

- Some of the cyber transforms insert new high-assurance components into the model
- The behavior of the component (its contract) is specified in AGREE
- SPLAT generates component implementations from their specifications
- SPLAT also generates a proof showing that the component implements its specification
- Other components (e.g., Attestation Manager) are pre-built pre-verified libraries
- Their implementations are essentially library functions that are added to the build, possibly with some configuration data from the model
- Code can be generated in the CakeML language which has a verified compiler

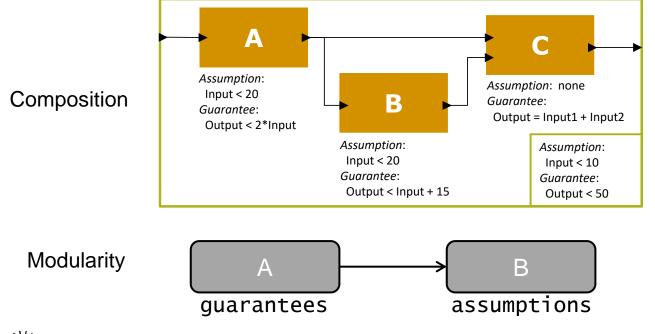


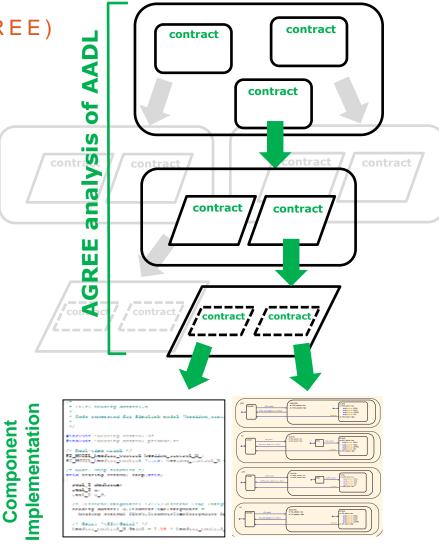


### ANALYZE SYSTEM BEHAVIOR

### ASSUME GUARANTEE REASONING ENVIRONMENT (AGREE)

- Contract-based compositional reasoning provides scalability
- Each component has a *contract* consisting of assumptions and guarantees
- The contract of a component abstracts the behavior of its implementation
- Contracts at each layer must be satisfied by contracts of its subcomponents
- Leaf component contracts must be satisfied by implementation

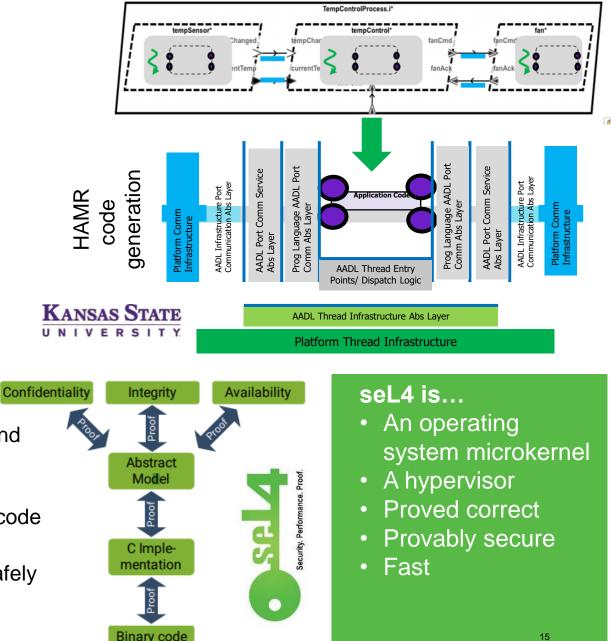




### SOFTWARE INFRASTRUCTURE

#### HAMR AND SEL4

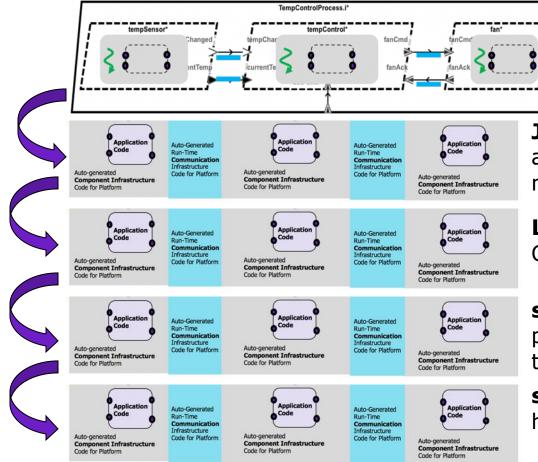
- HAMR is a multi-stage translation architecture to address CASE goals of component migration between platforms and information flow control
- Semantic consistency from model to execution
- Ensures model-level analysis applies to deployed code
- Same computational model across different platforms
- Build for multiple target platforms:
  - seL4 / Linux / Virtual Machine
  - Build for workstation / emulator / embedded platform
- seL4 microkernel guarantees partitioning of components and communication, backed by computer-checked proofs
- seL4 guarantees no infiltration, exfiltration, eavesdropping, interference, and provides fault containment for untrusted code
- Ensures soundness of the MBSE design process components can be analyzed separately and composed safely





### HAMR SUPPORTS MULTIPLE LANGUAGE/ PLATFORM COMBINATIONS

The flexibility of being able to easily shift between different platforms was quite useful as the team experimented with building the Phase 2 Experimental Platform assessment deliverable.



**AADL / OSATE** – design model, types, perform analyses

**JVM/Slang** – data types, port constraints, basic aspects of application logic, initial unit testing – some mocked up components, many useful visualizations

**Linux C** – compile Slang to C, or manually code C, and debug C implementation, VMs mocked up

**seL4 C / Qemu** – C application code easily ports to seL4 native components, add in VMs, test/simulate/debug in Qemu

**seL4 C / board** – seL4 build shifted to actual hardware for final testing and deployment



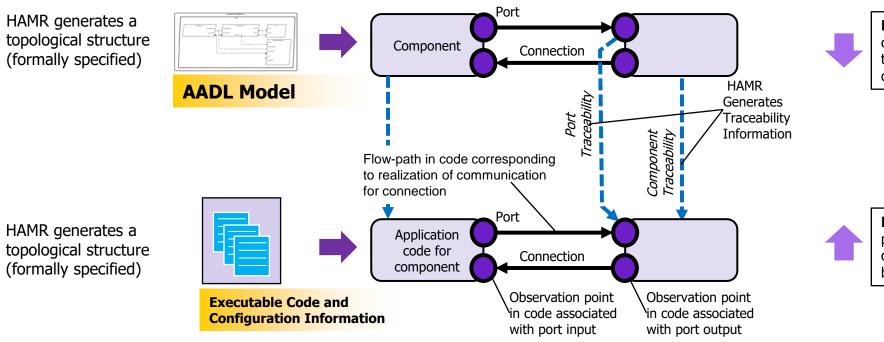
#### different programming languages Layer Port Layer Port Comm Abs Layer AADL Layei AADL AADL Infrastructure Communication Abs AADL Infrastructure Communication Abs Comm ADL Port Comm Service Abs Laye Abs Layer Application Platform Comm Abs Language Comm Language cture . . . Code Infrastructure . . . Comm Port nfrastru Platform ervice Port Plog Pbrt ( 60 communication AADL Thread Entry 5 APIs Points/ Dispatch Unit hamr\_SW\_WaypointPlanManagerService\_thr\_Impl\_Impl\_timeTriggered\_( Logic ́Т2 STACK FRAME hamr SW WaypointPlanManagerService thr Impl Impl this) { AADL Thread Infrastructure Abs Layer bool dataReceived = false; Т3 size t numBits = 0; uint8 t payload[MAX PAYLOAD]; Platform Thread Infrastructure dataReceived = api get ReturnHome hamr SW WaypointPlanManagerService thr Impl Impl (this); if (dataReceived) { Threading APIs returnHome = true; if (returnHome || automationResponse != NULL) { dataReceived = api get AirVehicleState hamr SW WaypointPlanManagerService thr Impl Impl ( this, &numBits, payload); if (dataReceived) air vehicle state in event data receive handler(this, payload); ollins Aerospace 17

### HAMR ABSTRACTION LAYERS

By changing the implementation of these layers, we can easily switch to **different platforms** or

### HAMR CORRESPONDENCE PROOF

- All information flows in AADL model are accurately preserved in HAMR generated code
- Connects AADL information flow analysis to seL4 security proofs



**FlowPreservation** (formal SMT spec): For every connection between two components in AADL, there is a flow path in the source code between code artifacts associated with the ports.

**NoNewFlows** (formal SMT spec): For every flow path between two components in the source code, there is a connection in the AADL model between corresponding ports.



### END-TO-END INTEGRATED FORMAL VERIFICATION

system properti			
architecture prope	architecture properties (AGREE)		
high-assurance components	legacy components		
HAMR correspo	assurance case		
CAmkES trans			
seL4 initializ			
seL4 p			



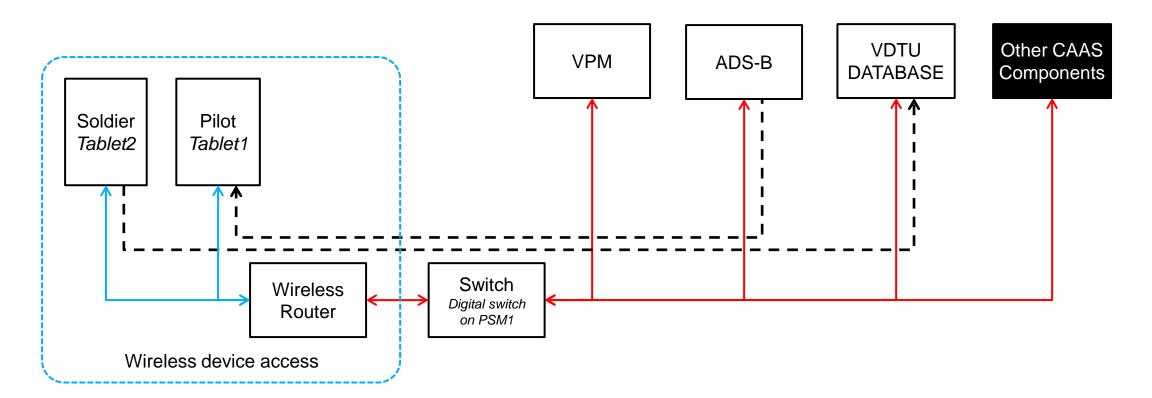
# CASE FINAL DEMO

#### COLLINS CUSTOMER EXPERIENCE CENTER HUNTSVILLE AL DECEMBER 2021

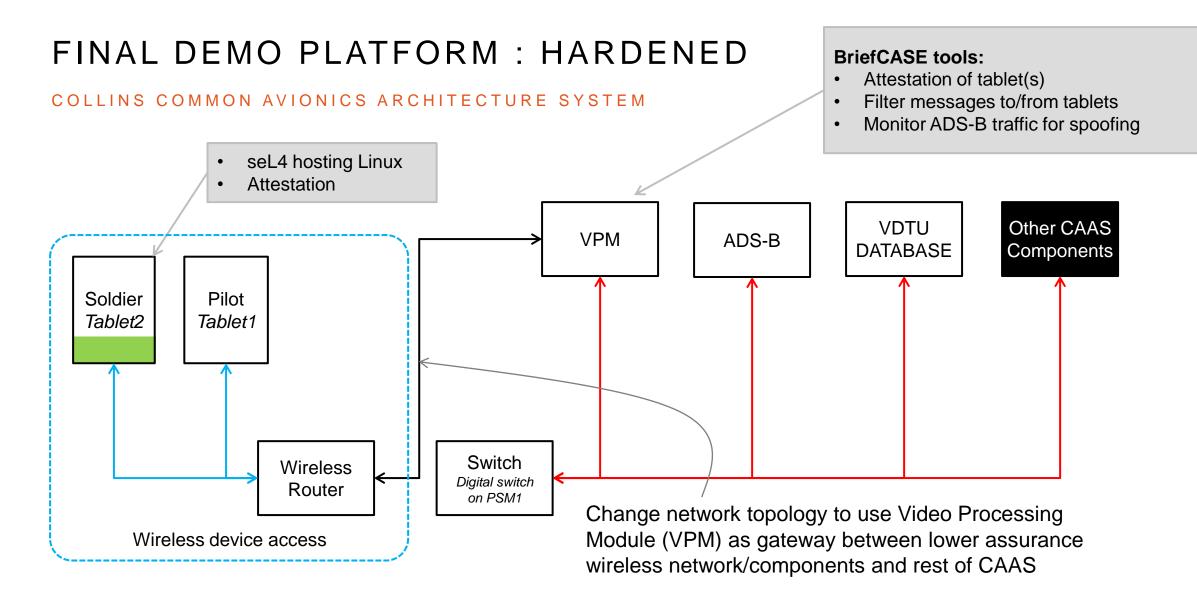
### FINAL DEMO PLATFORM : BASELINE

#### COLLINS COMMON AVIONICS ARCHITECTURE SYSTEM (CAAS)

• Goal : Extend (securely) to add wireless connectivity

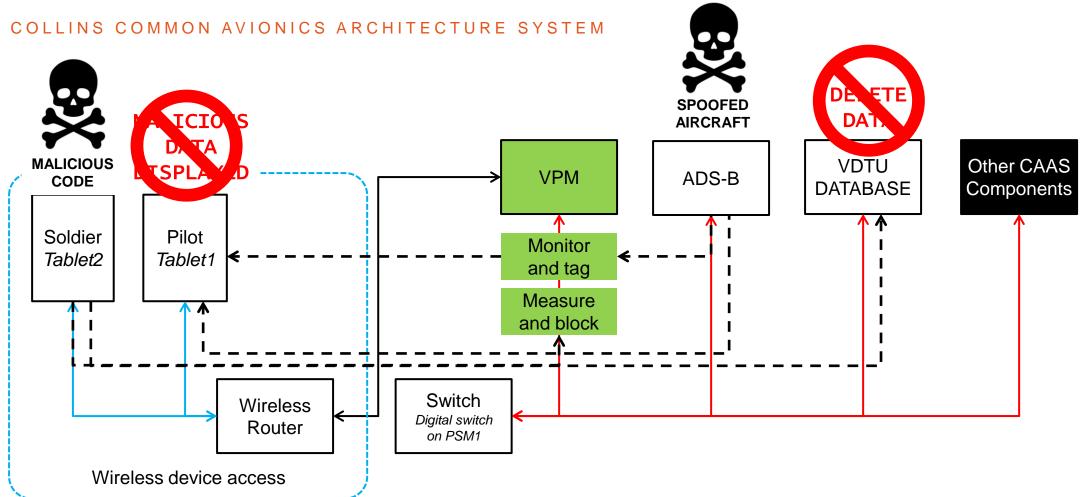








### **DEMO PLATFORM : ATTACKS**





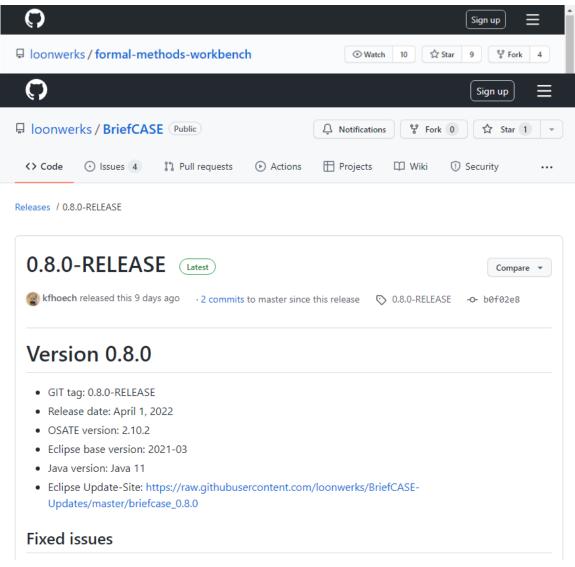
### OPEN-SOURCE SOFTWARE TOOL DISTRIBUTION

Tool source code resides in several public GitHub repositories

#### https://github.com/loonwerks/CASE-Final

also {/BriefCASE, /splat, /AGREE, /Resolute, /jkind}
https://github.com/ku-sldg
https://github.com/seL4
https://github.com/CakeML/cakeml
https://github.com/sireum

- Integrated OSATE/AADL tools and plugins
- Vagrant VM
  - Provides automatic, consistent, and reproducible provisioning of VM and native environments for developing and testing all CASE tools
- Documentation
  - Workflow example tutorial and models
  - User Guide
  - Videos, publications
- Overview
  - http://loonwerks.com/projects/case.html





### CYBER-ASSURED SYSTEMS ENGINEERING AT SCALE



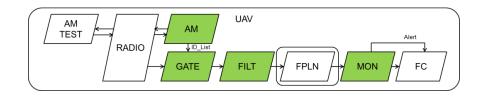
Also available at: https://loonwerks.com/publications/cofer2022secpriv.html



### BRIEFCASE TUTORIAL

#### Come to our Bootcamp session!

- Learn to use the BriefCASE tools
- Address cyber-resiliency requirements on a small example, analyze properties, generate code, create assurance argument, build and run system on seL4 (in QEMU)
- VM with all tools, models, and instructions
- Get it from Darren or Isaac, or download from github before the session



Day 4 (Bootca	mp, on-site participant	ts only) 13 October 2022			
9:00 - 10:00	Bootcamp	seL4: from zero to hello world Ihor Kuz, Kry10			
Break					
10:15 - 10:45	Bootcamp	CAmkES Sebastian Eckl, HENSOLDT Cyber			
10:45 - 12:00	Bootcamp	TRENTOS Sebastian Eckl, HENSOLDT Cyber			
Break					
13:00 - 14:30	Bootcamp	The seL4 Core Platform (seL4CP) Ivan Velickovic & Peter Chubb, UNSW			
14:30 - 15:15	Bootcamp	DornerWorks' VM Composer: the easy button for virtualized seL4-based systems Chris Guikema & Robbie VanVossen, Dornerworks			
Break					
15:30 - 17:00	Bootcamp	BriefCASE tutorial Isaac Amundson & Darren Cofer, Collins Aerospace			
17:00 - 17:05	Plenary	Concluding remarks			



### QUESTIONS?



