Cameo Enterprise Architecture Synchronized with Software Integration

MBSE Requirements to Integrated Code

APPROVED FOR PUBLIC RELEASE

11/13/2024

Agenda

- Software Integration - Problem Definition
- Context Setting
- Proposed Solution
- Demonstration
- Roadmap
- Open Discussion

The Problem Space

- **3-5 years – Average time to deploy new software (capability) to a defense system**

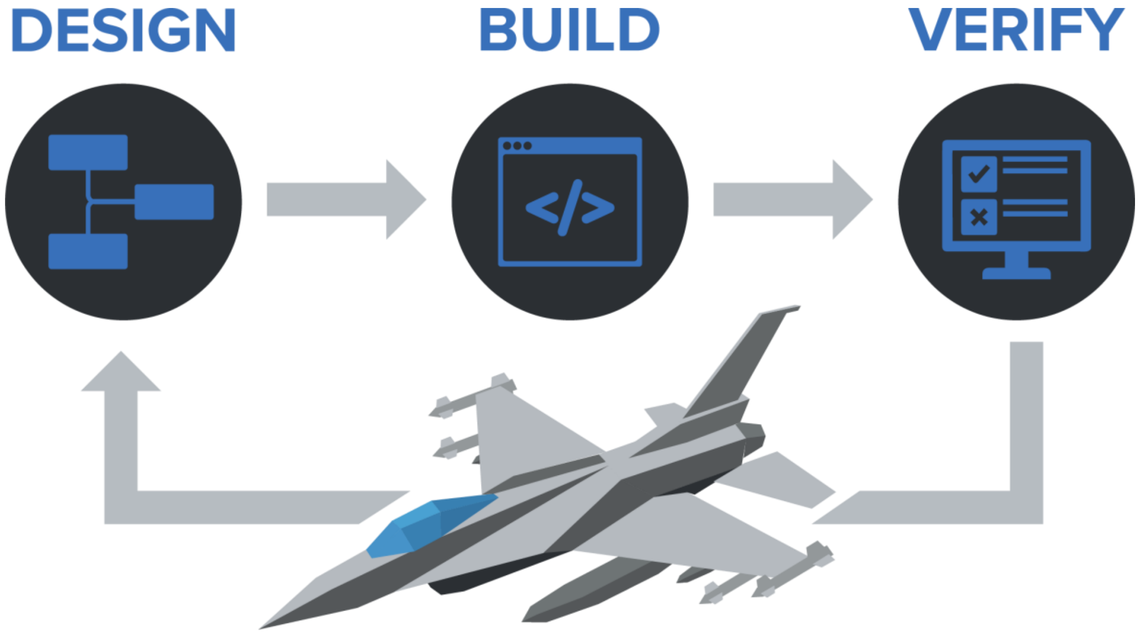
Missions demand faster capability updates

Changes require manual tasks

The Integration Challenge

Digital Engineering

- Ideas are “digitally born”
- Workflows and artifacts need to be ”digitally managed and stored”
  - – Collaborative environments
  - – Single source of truth

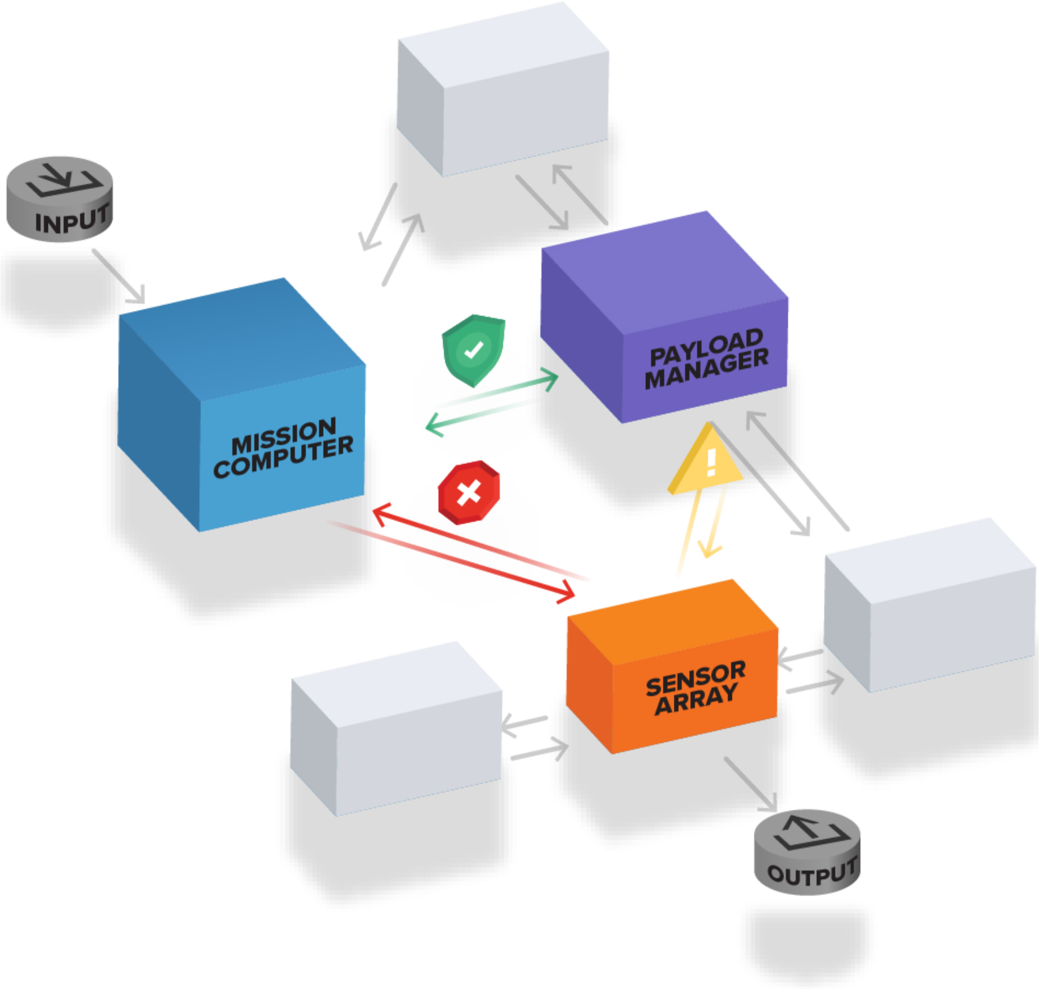


APPROVED FOR PUBLIC RELEASE

Model-Based Systems Engineering

NDIA

- System Architecture
  - Manage Complexity
  - Reduce Ambiguity
  - Embrace Creativity
- Collaborate with Stakeholders
- Shrink Schedules
- Manage throughout a system's Lifecycle



APPROVED FOR PUBLIC RELEASE

Software Integration

MOSA – Modular Open Systems Approach

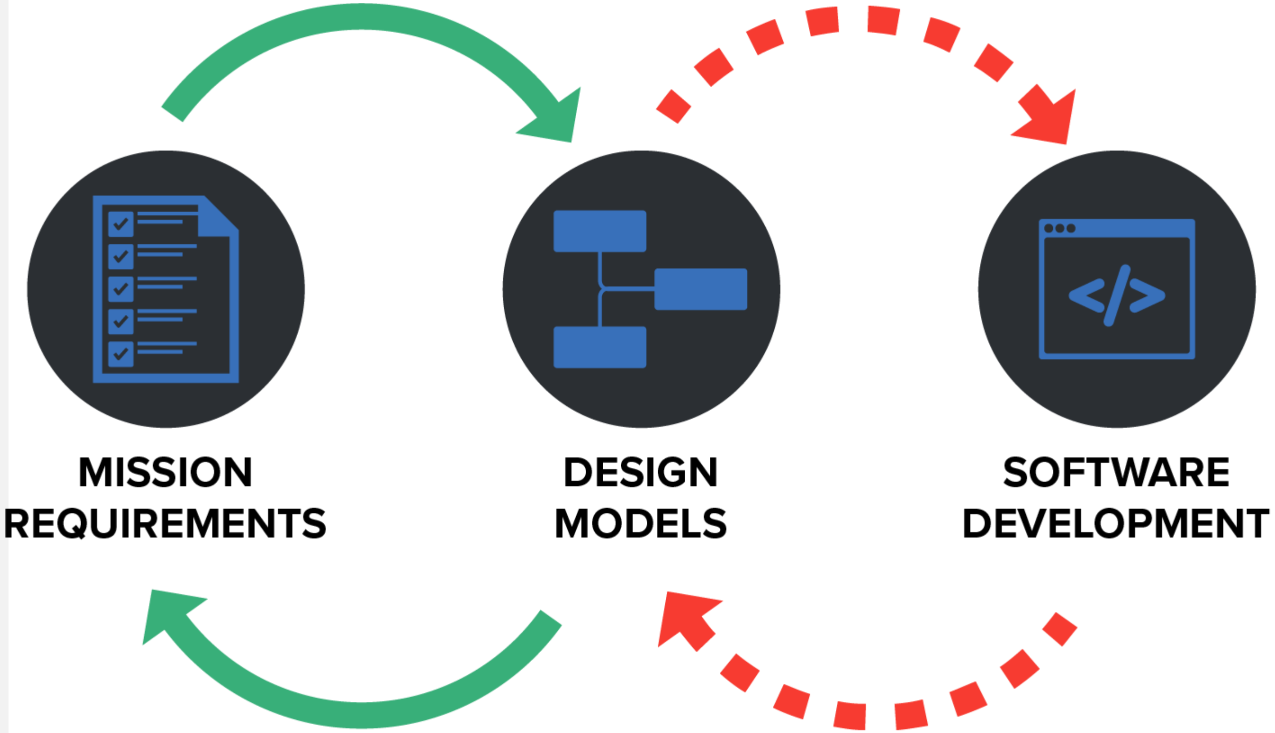
- Technical & Business strategy for designing affordable and adaptable systems
  - Software-defined interface syntax & properties for passing values among components
- Enabling Standards
  - Support interoperability, portability, and scalability
  - Unique architecture, data models, and abstractions

https://www.dsp.dla.mil/Programs/MOSA/

APPROVED FOR PUBLIC RELEASE

Gaps between Software & Systems

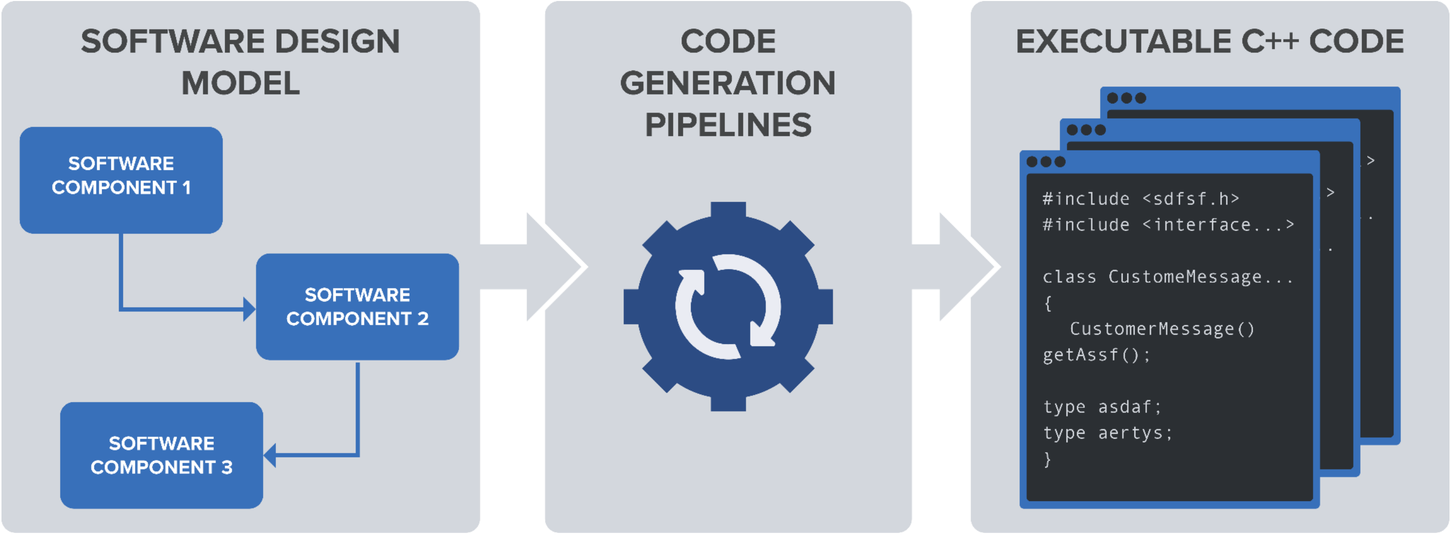
- System Architectures capture behavior among components
- …but lack traceability between Software Implementation and System Requirements
  - Negative impact on schedules
  - Cost overruns
  - Software Engineers misinterpret Requirements



APPROVED FOR PUBLIC RELEASE

Generating Code from Digital Models

- Model-Based approach to Software Integration
- Generate APIs from Digital Engineering models
  - Assurance
  - Verification
  - Interoperability
- Well defined Integration Point for rest of the system



```
#include <sdfsf.h>
#include <interface...>

class CustomeMessage...
{
   CustomerMessage()
getAssf();

type asdaf;
type aertys;
}
```

APPROVED FOR PUBLIC RELEASE

Artisan vs Automation

> “Digital Engineering must achieve a measure of authoritative virtualization that replaces, *automates*, or truncates formerly real-world activities”
>
> Dr. Will Roper – *Bending the Spoon*, 2021

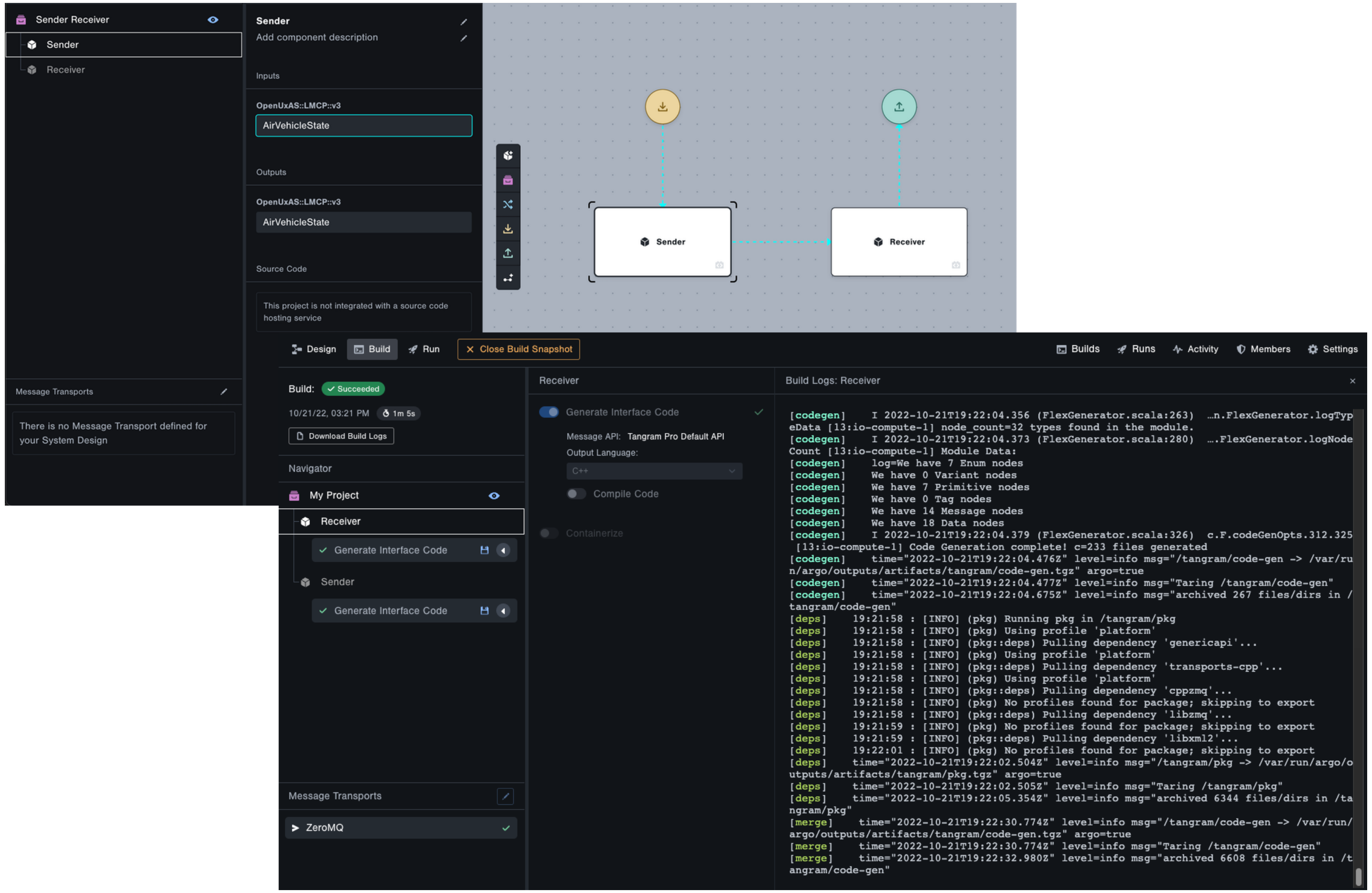
Tangram Pro

Model Based Software Integration Tool

NDIA

- Adopt SysML-like practices to Software Integration
- Manage Software Components
- Define Messages
- Message Transforms
- Generate API
- Automatic Code Generation

TANGRAMPRO™



Sender

Add component description

Inputs

OpenUxAS::LMCP::v3

AirVehicleState

Outputs

OpenUxAS::LMCP::v3

AirVehicleState

Source Code

This project is not integrated with a source code hosting service

Message Transports

There is no Message Transport defined for your System Design

Build: ✓ Succeeded

10/21/22, 03:21 PM   1m 5s

My Project

Receiver

✓ Generate Interface Code

Sender

✓ Generate Interface Code

Message Transports

ZeroMQ ✓

Receiver

Generate Interface Code ✓

Message API: Tangram Pro Default API

Output Language: C++

Compile Code

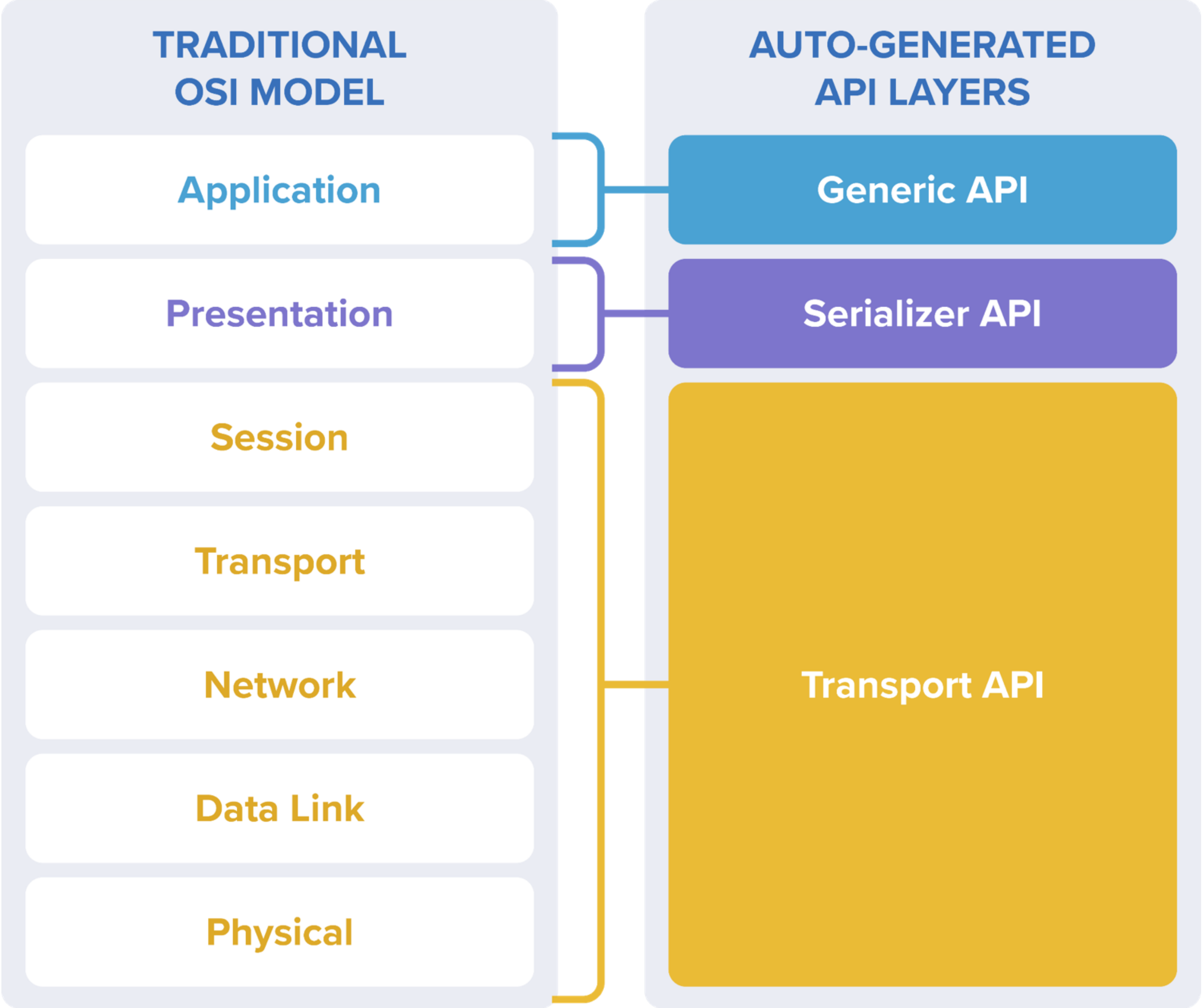
Build Logs: Receiver

```
[codegen]    I 2022-10-21T19:22:04.356 (FlexGenerator.scala:263)  …n.FlexGenerator.logTyp
eData [13:io-compute-1] node_count=32 types found in the module.
[codegen]    I 2022-10-21T19:22:04.373 (FlexGenerator.scala:280)  ….FlexGenerator.logNode
Count [13:io-compute-1] Module Data:
[codegen]    log=We have 7 Enum nodes
[codegen]    We have 0 Variant nodes
[codegen]    We have 7 Primitive nodes
[codegen]    We have 0 Tag nodes
[codegen]    We have 14 Message nodes
[codegen]    We have 18 Data nodes
[codegen]    I 2022-10-21T19:22:04.379 (FlexGenerator.scala:326)  c.F.codeGenOpts.312.325
 [13:io-compute-1] Code Generation complete! c=233 files generated
[codegen]    time="2022-10-21T19:22:04.476Z" level=info msg="/tangram/code-gen -> /var/ru
n/argo/outputs/artifacts/tangram/code-gen.tgz" argo=true
[codegen]    time="2022-10-21T19:22:04.477Z" level=info msg="Taring /tangram/code-gen"
[codegen]    time="2022-10-21T19:22:04.675Z" level=info msg="archived 267 files/dirs in /
tangram/code-gen"
[deps]    19:21:58 : [INFO] (pkg) Running pkg in /tangram/pkg
[deps]    19:21:58 : [INFO] (pkg) Using profile 'platform'
[deps]    19:21:58 : [INFO] (pkg::deps) Pulling dependency 'genericapi'...
[deps]    19:21:58 : [INFO] (pkg) Using profile 'platform'
[deps]    19:21:58 : [INFO] (pkg::deps) Pulling dependency 'transports-cpp'...
[deps]    19:21:58 : [INFO] (pkg) Using profile 'platform'
[deps]    19:21:58 : [INFO] (pkg::deps) Pulling dependency 'cppzmq'...
[deps]    19:21:58 : [INFO] (pkg) No profiles found for package; skipping to export
[deps]    19:21:58 : [INFO] (pkg::deps) Pulling dependency 'libzmq'...
[deps]    19:21:59 : [INFO] (pkg) No profiles found for package; skipping to export
[deps]    19:21:59 : [INFO] (pkg::deps) Pulling dependency 'libxml2'...
[deps]    19:22:01 : [INFO] (pkg) No profiles found for package; skipping to export
[deps]    time="2022-10-21T19:22:02.504Z" level=info msg="/tangram/pkg -> /var/run/argo/o
utputs/artifacts/tangram/pkg.tgz" argo=true
[deps]    time="2022-10-21T19:22:02.505Z" level=info msg="Taring /tangram/pkg"
[deps]    time="2022-10-21T19:22:05.354Z" level=info msg="archived 6344 files/dirs in /ta
ngram/pkg"
[merge]    time="2022-10-21T19:22:30.774Z" level=info msg="/tangram/code-gen -> /var/run/
argo/outputs/artifacts/tangram/code-gen.tgz" argo=true
[merge]    time="2022-10-21T19:22:30.774Z" level=info msg="Taring /tangram/code-gen"
[merge]    time="2022-10-21T19:22:32.980Z" level=info msg="archived 6608 files/dirs in /t
angram/code-gen"
```

APPROVED FOR PUBLIC RELEASE

Tangram Pro Generated APIs

- Generated API provides entire communication libraries based on Data Models
- Modularity mapped to OSI model
- Allows integration at any layer in the stack

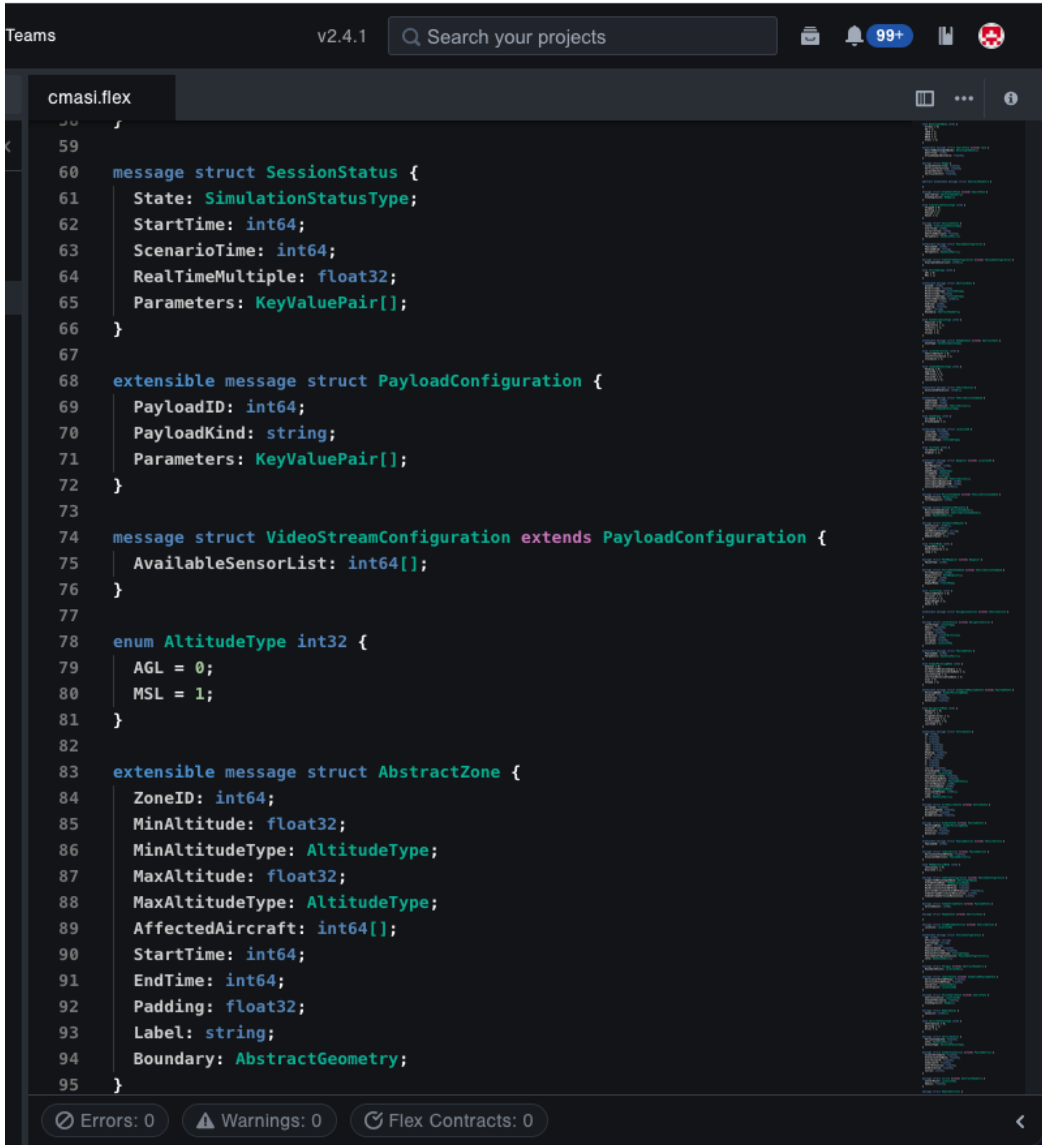


APPROVED FOR PUBLIC RELEASE

Tangram Pro

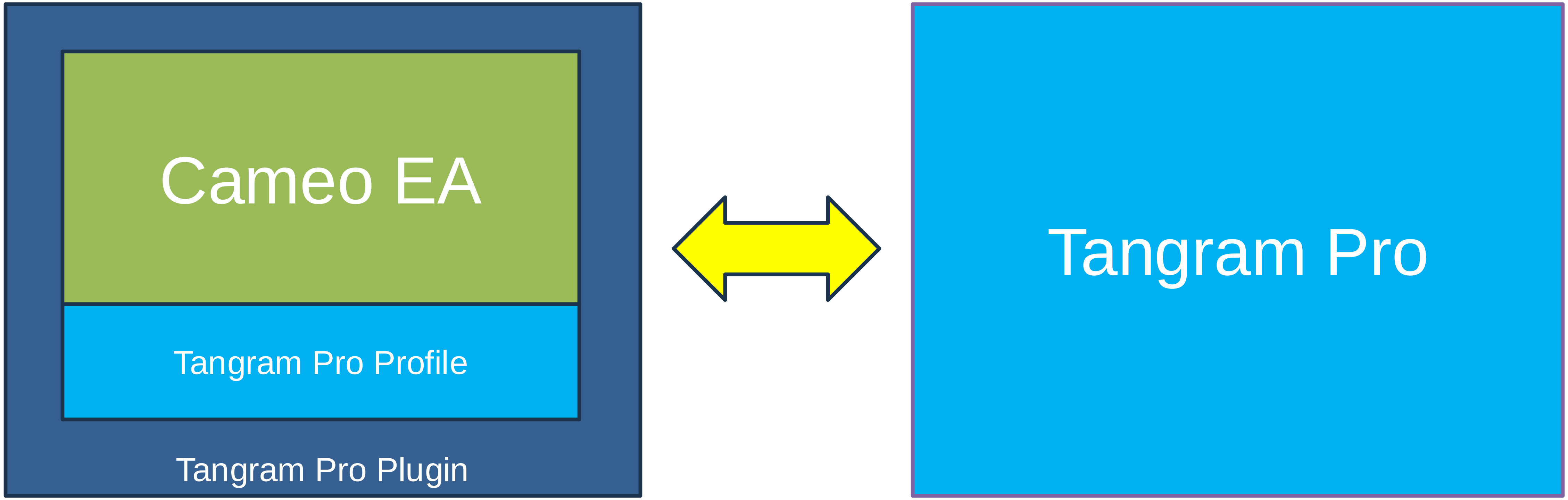
Flex Language

- Open Domain Specific Specification Language developed by Tangram Flex
- Familiar Syntax
- Import from IDL, XSD, Proto…
- Specifies Data Models & Relationships between them
  - Transform
  - Separation of Concerns
- Correctness



```
}

message struct SessionStatus {
  State: SimulationStatusType;
  StartTime: int64;
  ScenarioTime: int64;
  RealTimeMultiple: float32;
  Parameters: KeyValuePair[];
}

extensible message struct PayloadConfiguration {
  PayloadID: int64;
  PayloadKind: string;
  Parameters: KeyValuePair[];
}

message struct VideoStreamConfiguration extends PayloadConfiguration {
  AvailableSensorList: int64[];
}

enum AltitudeType int32 {
  AGL = 0;
  MSL = 1;
}

extensible message struct AbstractZone {
  ZoneID: int64;
  MinAltitude: float32;
  MinAltitudeType: AltitudeType;
  MaxAltitude: float32;
  MaxAltitudeType: AltitudeType;
  AffectedAircraft: int64[];
  StartTime: int64;
  EndTime: int64;
  Padding: float32;
  Label: string;
  Boundary: AbstractGeometry;
}
```

Cameo Integrated with Tangram Pro



APPROVED FOR PUBLIC RELEASE

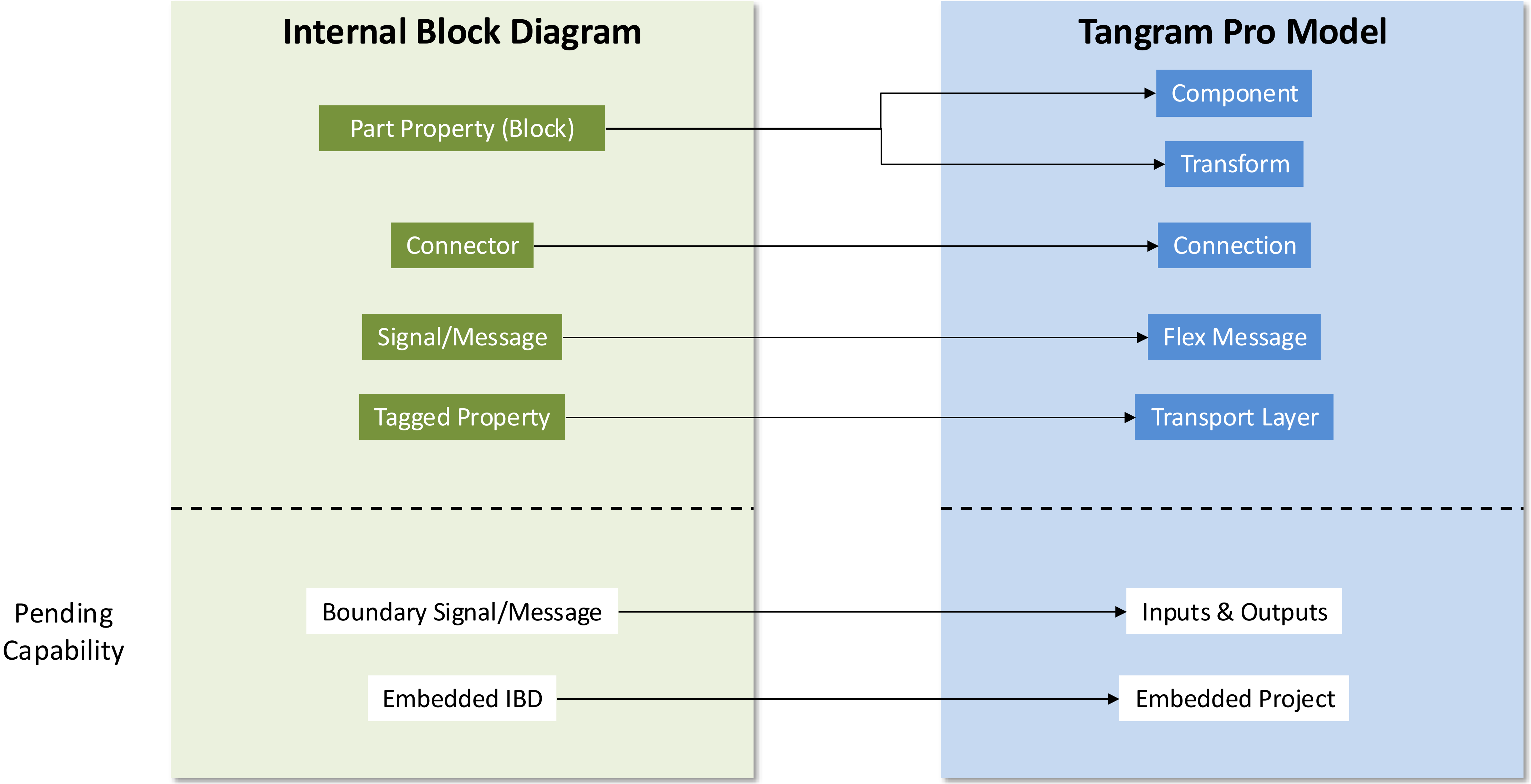
Value Proposition

- Maintain Authoritative Source of Truth
- Integrate with legacy and new capability
- Unite Systems and Software Engineering
- Save time and resources
- Eliminate barriers to innovation

Cameo – Tangram Pro Plugin & Profile

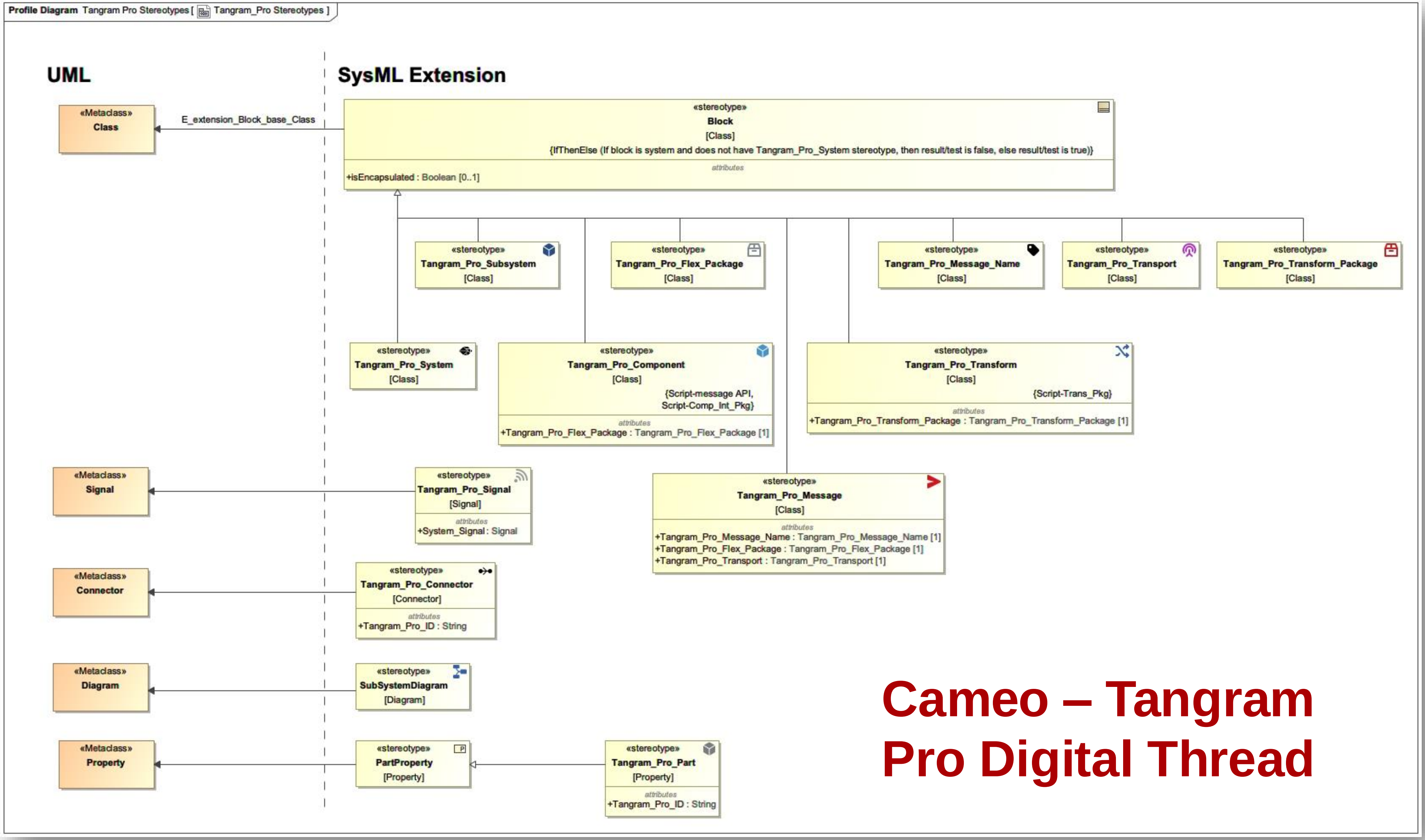
- Tangram Pro Plugin
  - Built referencing Cameo Developer Guide
    - https://docs.nomagic.com/display/MD2022xR2/Developer+Guide
  - Developed using IntelliJ
  - Referenced Cameo generated Javadoc
- Profile
  - Stereotypes matching SysML entities to a Tangram Pro model
  - Customizations for Derived Properties and Diagrams
  - Custom diagrams (IBD and BDD like)
  - Validation rules
  - Style guide and sample models
- Distributed as a Cameo Resource Distribution File (.rdzip)

Cameo – Tangram Pro Digital Thread



APPROVED FOR PUBLIC RELEASE

Cameo – Tangram Pro Digital Thread



Profile Diagram Tangram Pro Stereotypes [Tangram_Pro Stereotypes]

UML

SysML Extension

«Metaclass» Class

E_extension_Block_base_Class

«stereotype» Block [Class]
{IfThenElse (If block is system and does not have Tangram_Pro_System stereotype, then result/test is false, else result/test is true)}
attributes
+isEncapsulated : Boolean [0..1]

«stereotype» Tangram_Pro_Subsystem [Class]

«stereotype» Tangram_Pro_Flex_Package [Class]

«stereotype» Tangram_Pro_Message_Name [Class]

«stereotype» Tangram_Pro_Transport [Class]

«stereotype» Tangram_Pro_Transform_Package [Class]

«stereotype» Tangram_Pro_System [Class]

«stereotype» Tangram_Pro_Component [Class]
{Script-message API, Script-Comp_Int_Pkg}
attributes
+Tangram_Pro_Flex_Package : Tangram_Pro_Flex_Package [1]

«stereotype» Tangram_Pro_Transform [Class]
{Script-Trans_Pkg}
attributes
+Tangram_Pro_Transform_Package : Tangram_Pro_Transform_Package [1]

«Metaclass» Signal

«stereotype» Tangram_Pro_Signal [Signal]
attributes
+System_Signal : Signal

«stereotype» Tangram_Pro_Message [Class]
attributes
+Tangram_Pro_Message_Name : Tangram_Pro_Message_Name [1]
+Tangram_Pro_Flex_Package : Tangram_Pro_Flex_Package [1]
+Tangram_Pro_Transport : Tangram_Pro_Transport [1]

«Metaclass» Connector

«stereotype» Tangram_Pro_Connector [Connector]
attributes
+Tangram_Pro_ID : String

«Metaclass» Diagram

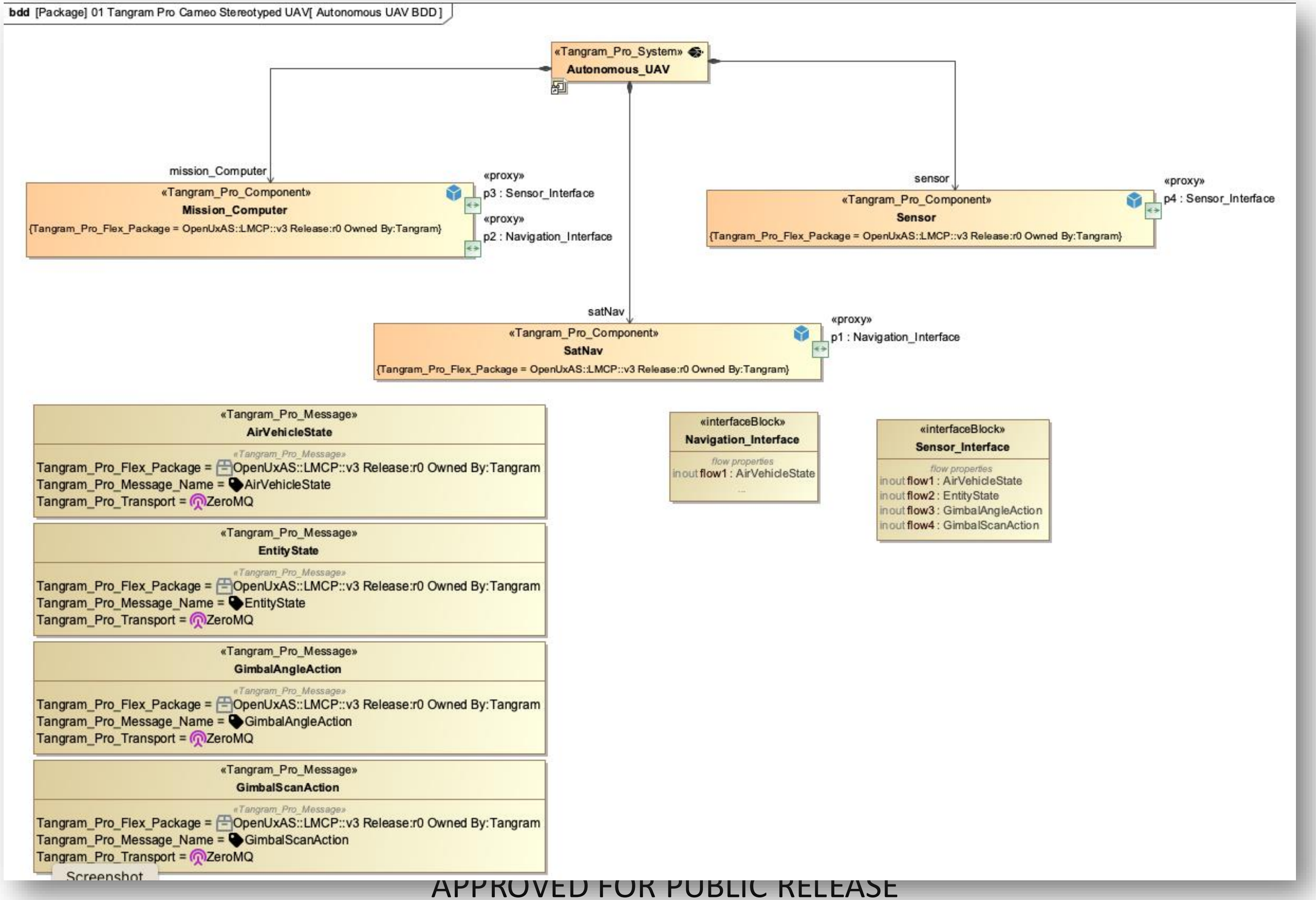
«stereotype» SubSystemDiagram [Diagram]

«Metaclass» Property

«stereotype» PartProperty [Property]

«stereotype» Tangram_Pro_Part [Property]
attributes
+Tangram_Pro_ID : String

Example BDD



bdd [Package] 01 Tangram Pro Cameo Stereotyped UAV[Autonomous UAV BDD]

«Tangram_Pro_System»
Autonomous_UAV

mission_Computer

«Tangram_Pro_Component»
Mission_Computer
{Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram}

«proxy»
p3 : Sensor_Interface

«proxy»
p2 : Navigation_Interface

sensor

«Tangram_Pro_Component»
Sensor
{Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram}

«proxy»
p4 : Sensor_Interface

satNav

«Tangram_Pro_Component»
SatNav
{Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram}

«proxy»
p1 : Navigation_Interface

«Tangram_Pro_Message»
AirVehicleState
«Tangram_Pro_Message»
Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram
Tangram_Pro_Message_Name = AirVehicleState
Tangram_Pro_Transport = ZeroMQ

«Tangram_Pro_Message»
EntityState
«Tangram_Pro_Message»
Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram
Tangram_Pro_Message_Name = EntityState
Tangram_Pro_Transport = ZeroMQ

«Tangram_Pro_Message»
GimbalAngleAction
«Tangram_Pro_Message»
Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram
Tangram_Pro_Message_Name = GimbalAngleAction
Tangram_Pro_Transport = ZeroMQ

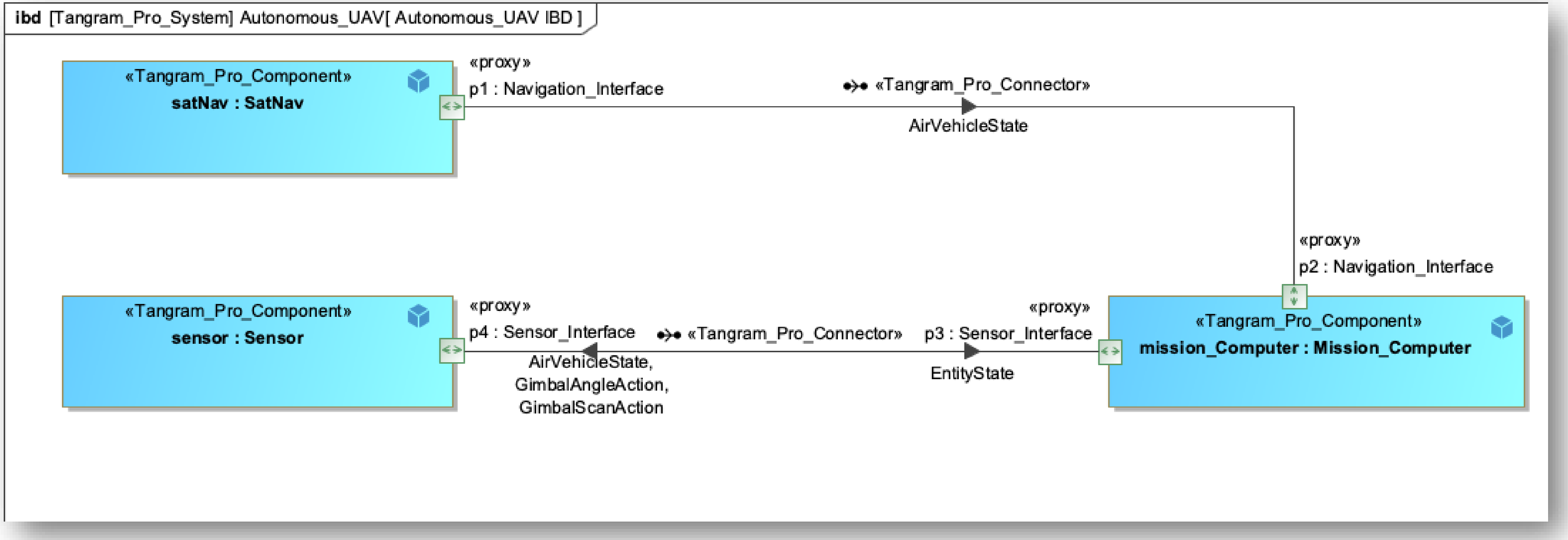
«Tangram_Pro_Message»
GimbalScanAction
«Tangram_Pro_Message»
Tangram_Pro_Flex_Package = OpenUxAS::LMCP::v3 Release:r0 Owned By:Tangram
Tangram_Pro_Message_Name = GimbalScanAction
Tangram_Pro_Transport = ZeroMQ

«interfaceBlock»
Navigation_Interface
flow properties
inout flow1 : AirVehicleState
...

«interfaceBlock»
Sensor_Interface
flow properties
inout flow1 : AirVehicleState
inout flow2 : EntityState
inout flow3 : GimbalAngleAction
inout flow4 : GimbalScanAction

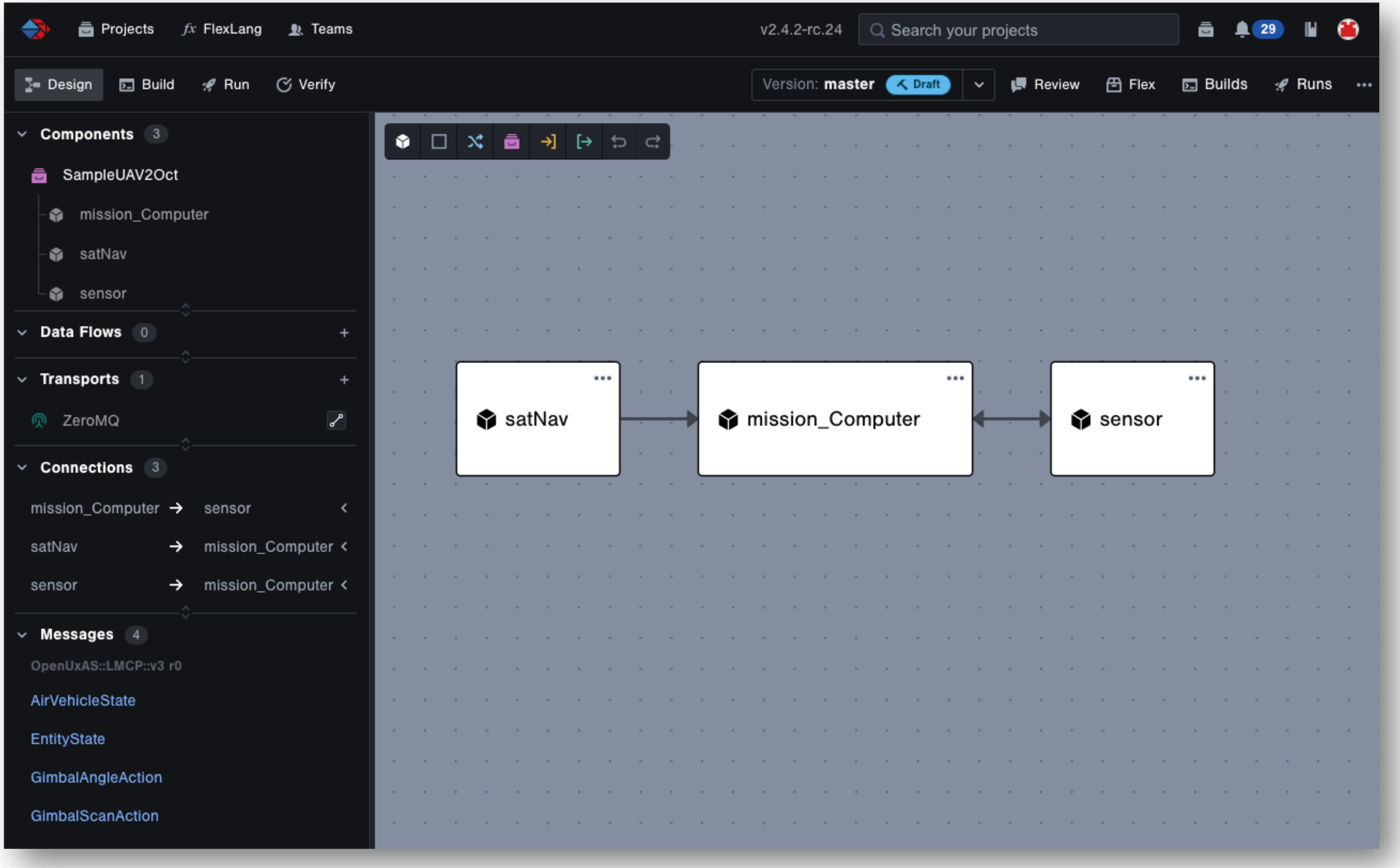
Screenshot

Example IBD



ibd [Tangram_Pro_System] Autonomous_UAV[Autonomous_UAV IBD]
«Tangram_Pro_Component»
satNav : SatNav
«proxy»
p1 : Navigation_Interface
«Tangram_Pro_Connector»
AirVehicleState
«proxy»
p2 : Navigation_Interface
«Tangram_Pro_Component»
sensor : Sensor
«proxy»
p4 : Sensor_Interface
«Tangram_Pro_Connector»
«proxy»
p3 : Sensor_Interface
AirVehicleState,
GimbalAngleAction,
GimbalScanAction
EntityState
«Tangram_Pro_Component»
mission_Computer : Mission_Computer

Resultant Tangram Pro Model



Components 3

SampleUAV2Oct
- mission_Computer
- satNav
- sensor

Data Flows 0

Transports 1

ZeroMQ

Connections 3

mission_Computer	→	sensor
satNav	→	mission_Computer
sensor	→	mission_Computer

Messages 4

OpenUxAS::LMCP::v3 r0

AirVehicleState

EntityState

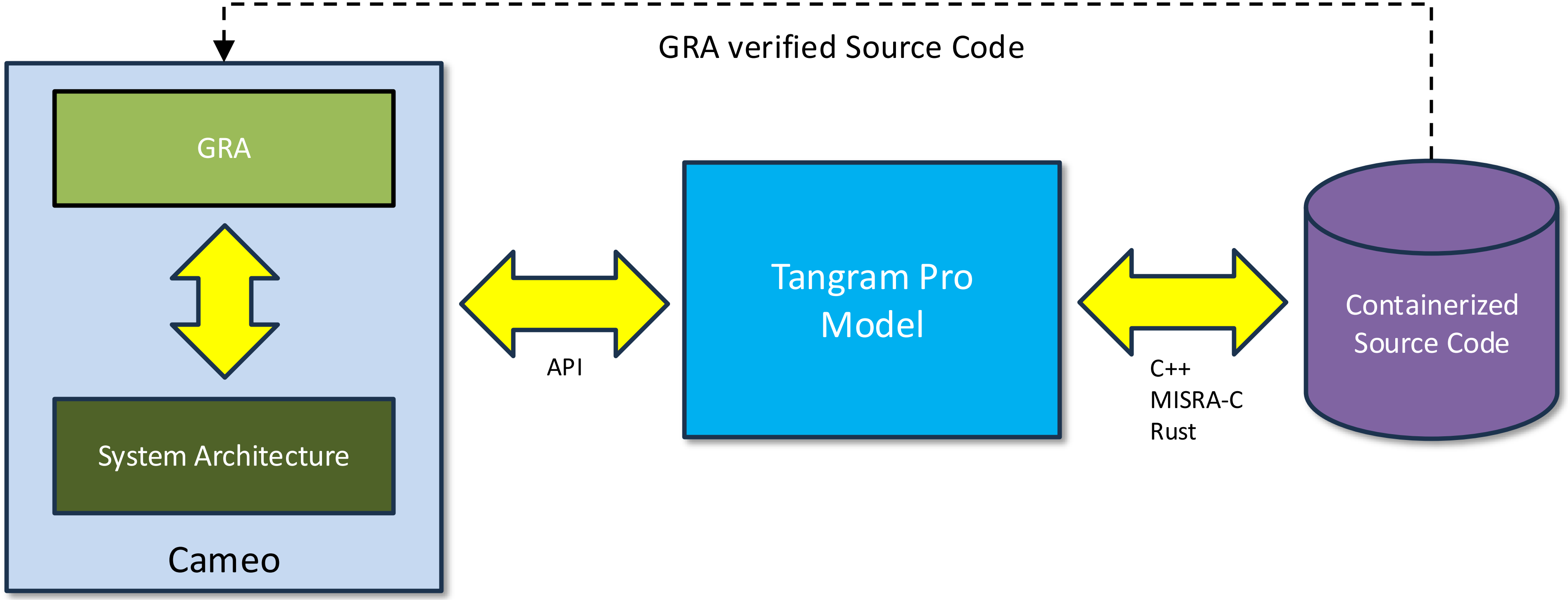
GimbalAngleAction

GimbalScanAction

APPROVED FOR PUBLIC RELEASE

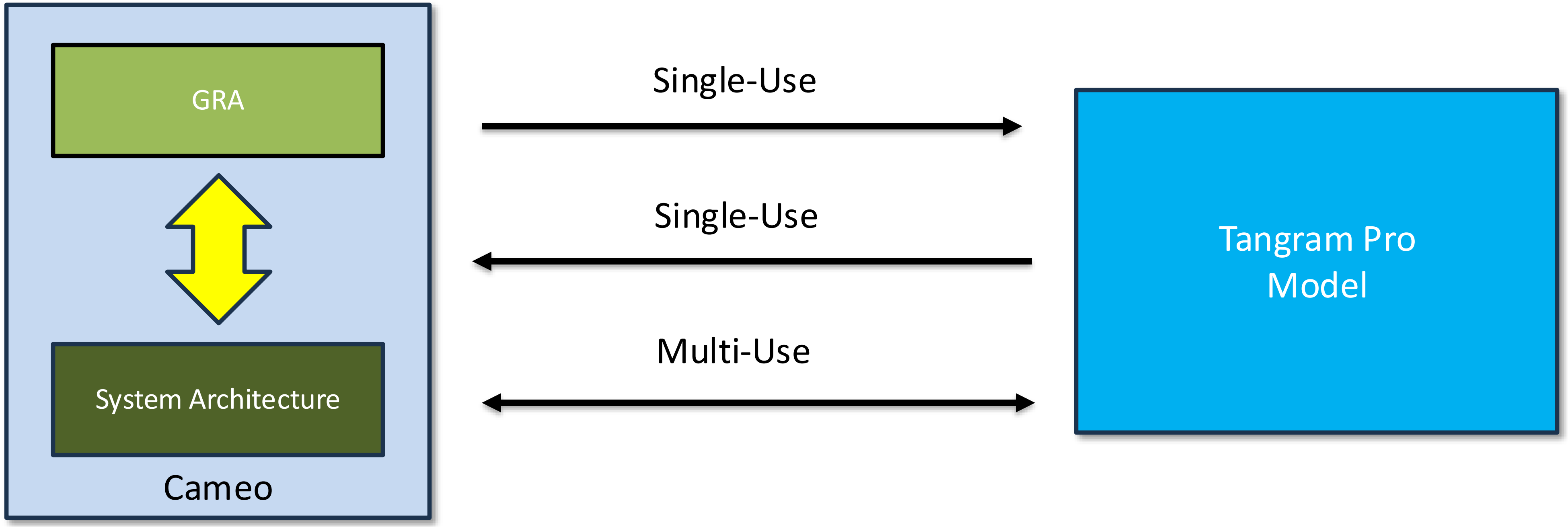
Cameo-Tangram Pro Use Case

- Government is defining Open Standards for various domains: Government Reference Architecture (GRA)
- System Architecture derived from GRA
- Software Integration Model (Tangram Pro) derived from System Architecture
- “Code Gen” pipeline traced to GRA



APPROVED FOR PUBLIC RELEASE

Cameo-Tangram Pro Scenarios



APPROVED FOR PUBLIC RELEASE

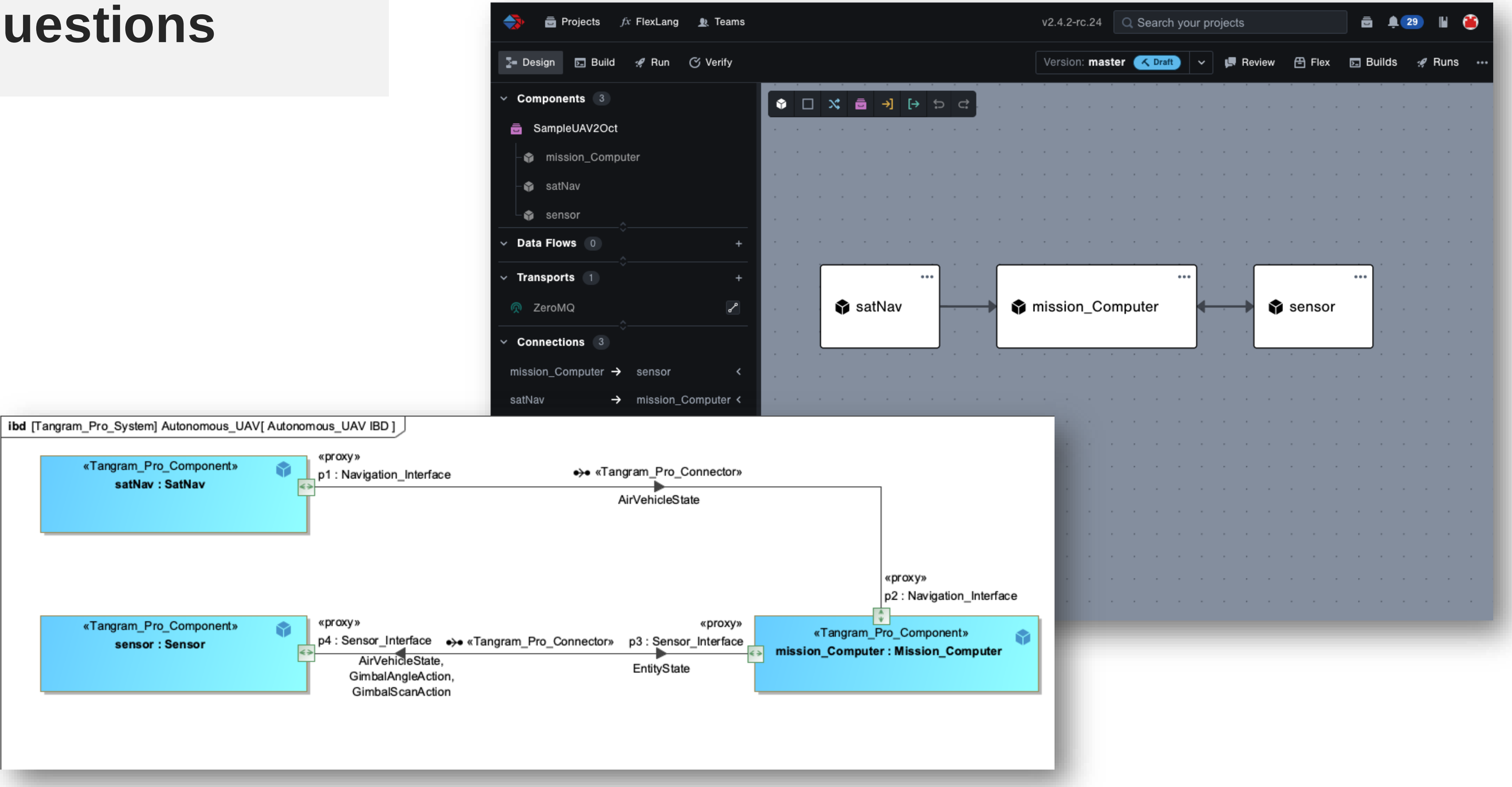
DEMONSTRATION

Roadmap

- Sync Cameo and Tangram Pro using Sequence Diagrams
- Embedded IBDs
- Boundary messages
- “Diffing” between Cameo and Tangram Pro models
- Integration with Application Code Development Tools
- Current testing with US Air Force System Architectures
  - WOSA: Enterprise Test Vehicle
  - ABMS
  - Big Iron

APPROVED FOR PUBLIC RELEASE

- # Questions



APPROVED FOR PUBLIC RELEASE

NDIA

Contact

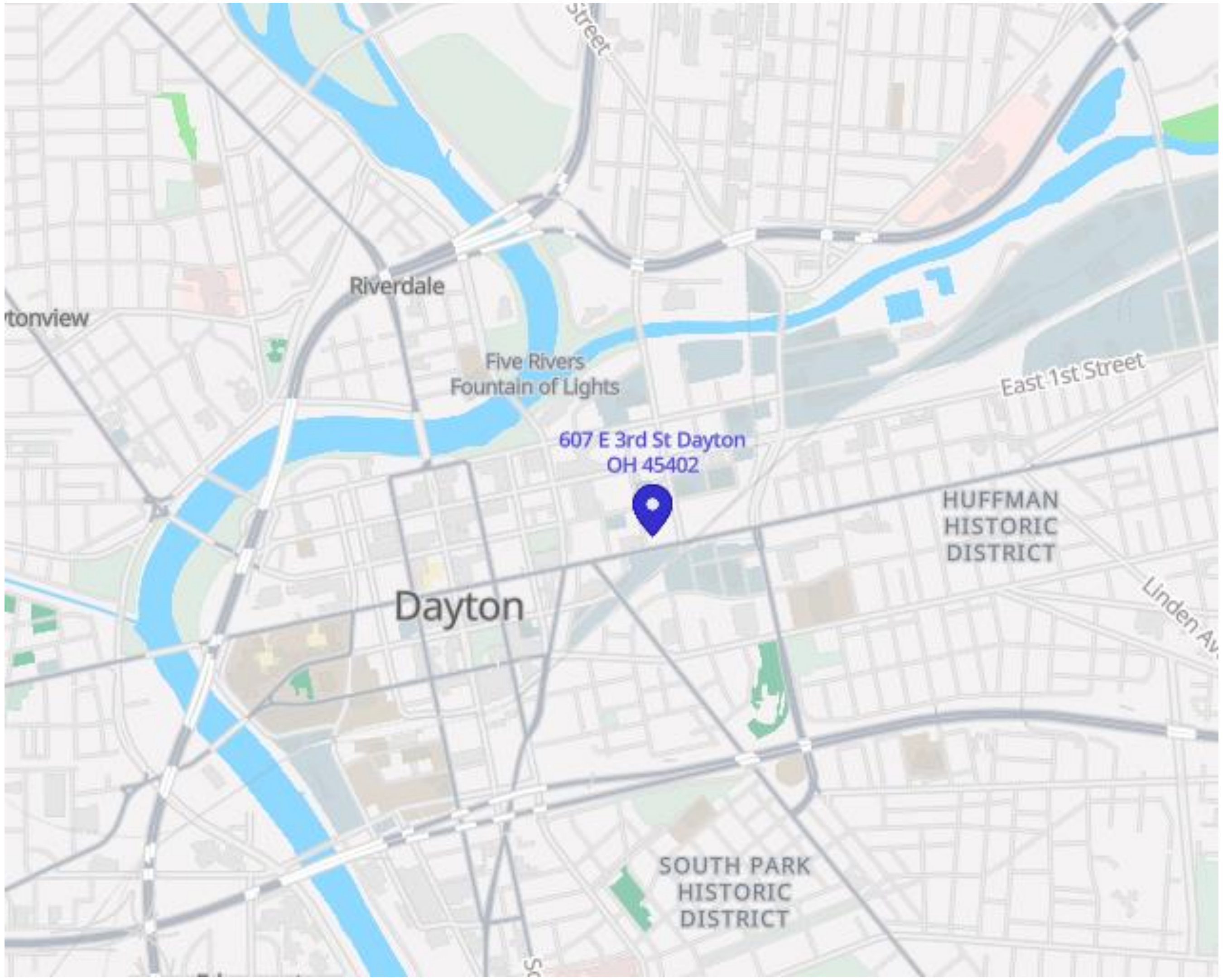
T.J. Hoke

Systems Engineer

TANGRAMFLEX®

tjhoke@tangramflex.com

937.416.5680



Riverdale
tonview
Five Rivers
Fountain of Lights
607 E 3rd St Dayton
OH 45402
East 1st Street
HUFFMAN
HISTORIC
DISTRICT
Dayton
Linden Av
SOUTH PARK
HISTORIC
DISTRICT
Street