Approved for Public Release; Distribution Unlimited.
Public Release Case Number 21-3597

DEWS Open Reference Architecture Development

Dr. Steven A. Davidson, MITRE

The Team: Saugat Sen, Alan R. Galgano, Ronald A Kacsmar, Eric M Drewniak, Grace B Unangst, Howard Lazoff , Zachary J Tretler, Leah E Balkin, Davin S. O'Neil, Keegan Merkert, Conor Michael Mahoney , William B. Donaldson, Christopher W Shaffer

This technical data deliverable was developed using contract funds under Basic Contract No. W56KGU-18-D-0004

December 8, 2021

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

Key Take-Aways

An open Reference Architecture – as an Open Systems Architecture – is a very effective means of applying MOSA

There is an art and a science of generating an open RefArch that involves

1. Understanding the ”care-abouts” and applying them to modularity
2. Defining the modularity by aggregating functions
3. Using the modularity to define the interfaces

The time is right to apply MOSA to directed energy weapon systems (DEWS) – and that’s what we’ve done

MITRE

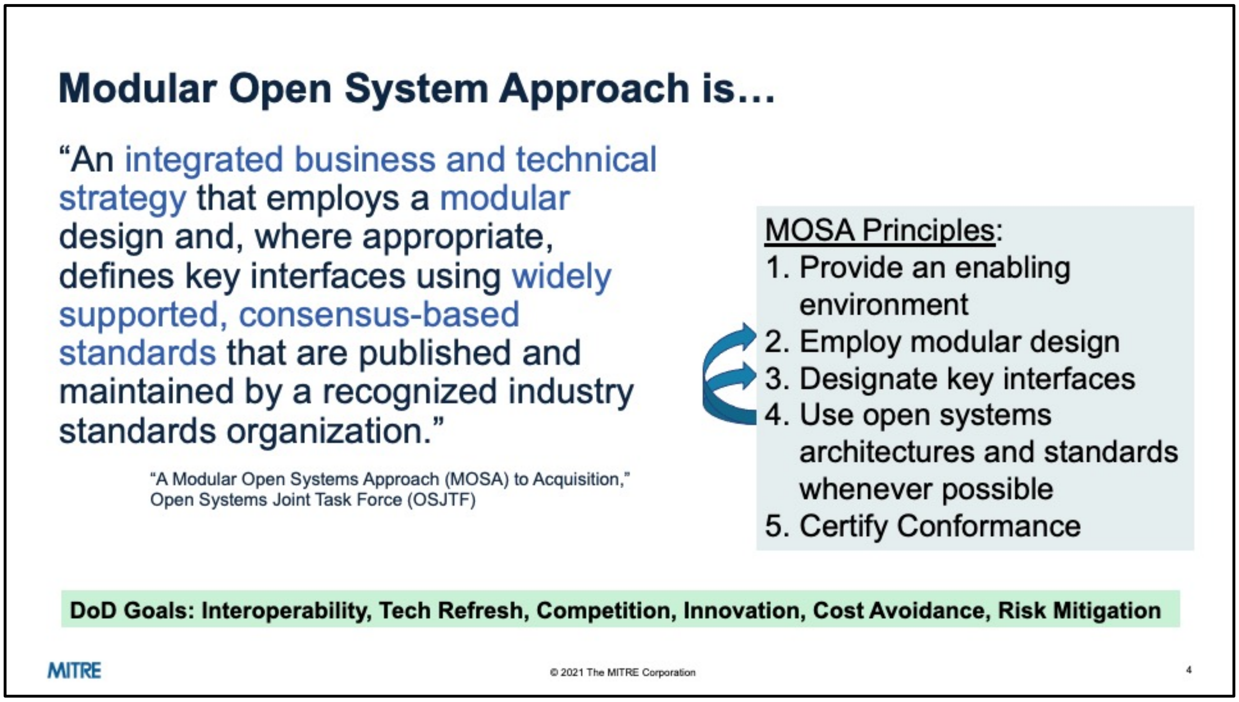
Table of Contents

MITRE

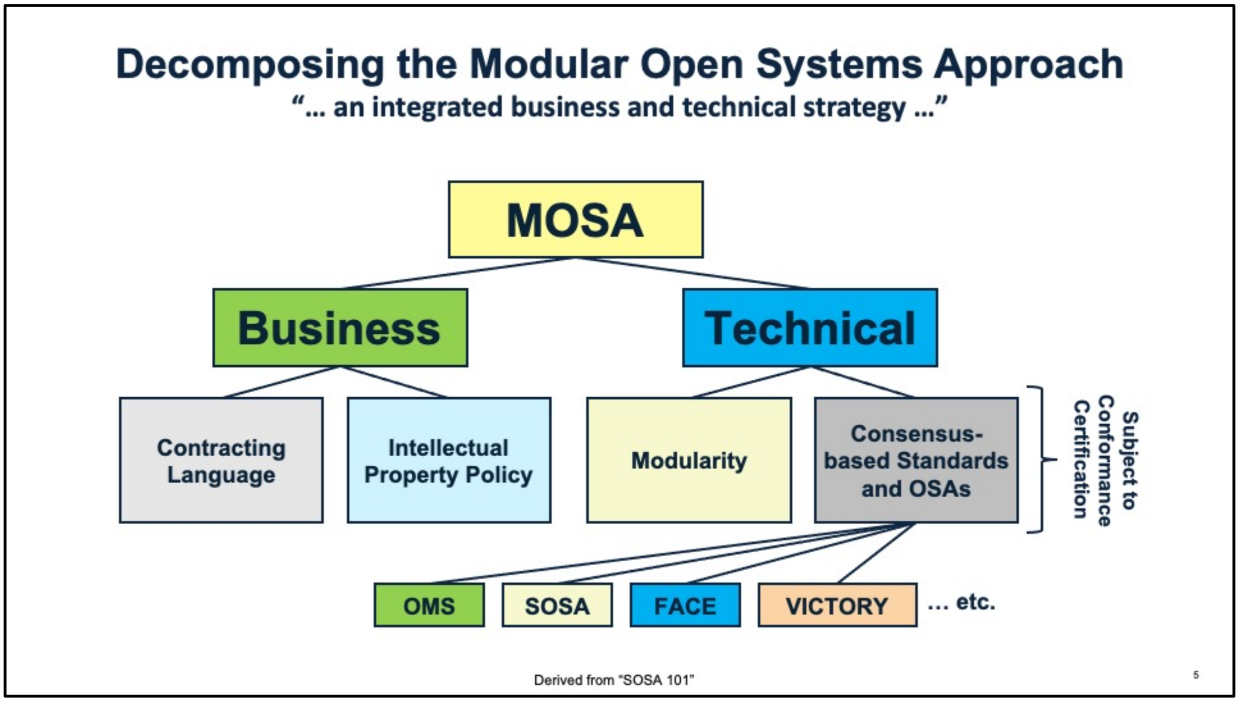
3

In the course of this presentation, I am going to answer these three questions: Why does MOSA Matter? What the heck is MOSA, anyway? And How is MOSA being applied to Directed Energy systems?

Let's begin with "why..."

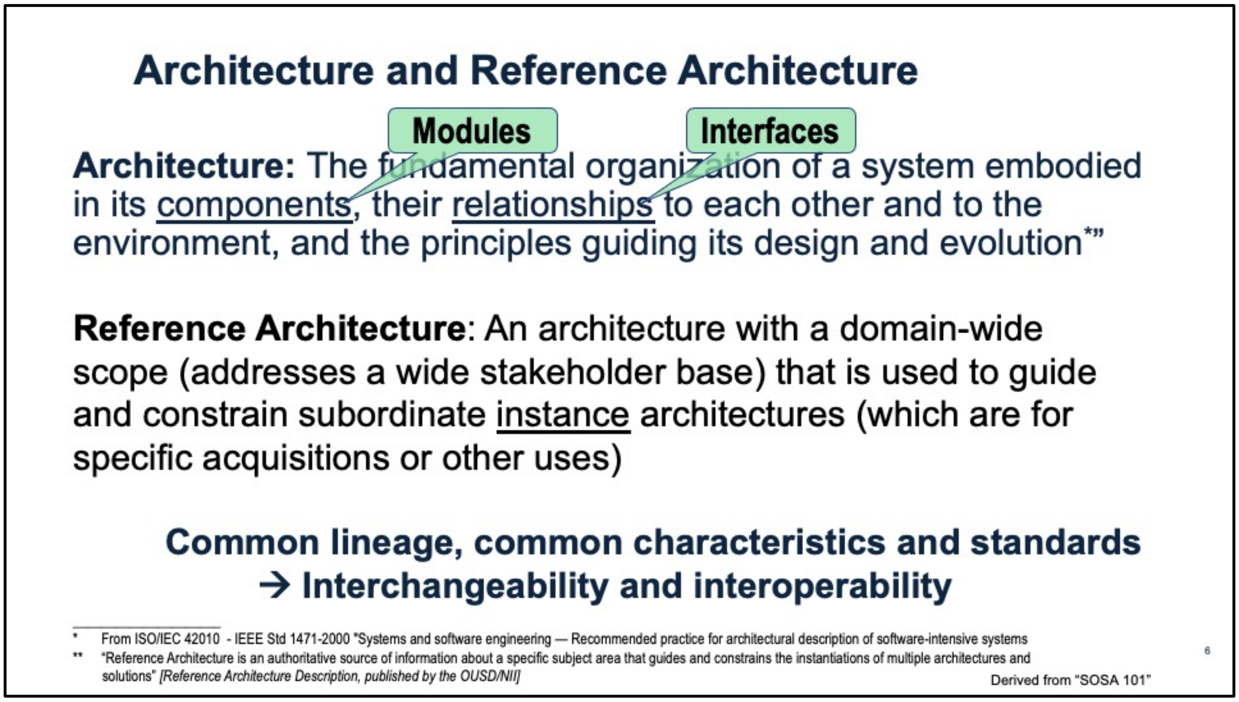


MOSA is an <u>integrated</u> combination of business/acquisition and technical content. There are officially five MOSA principles, but I would argue that if you use open systems architectures, you get #2 and #3 automatically



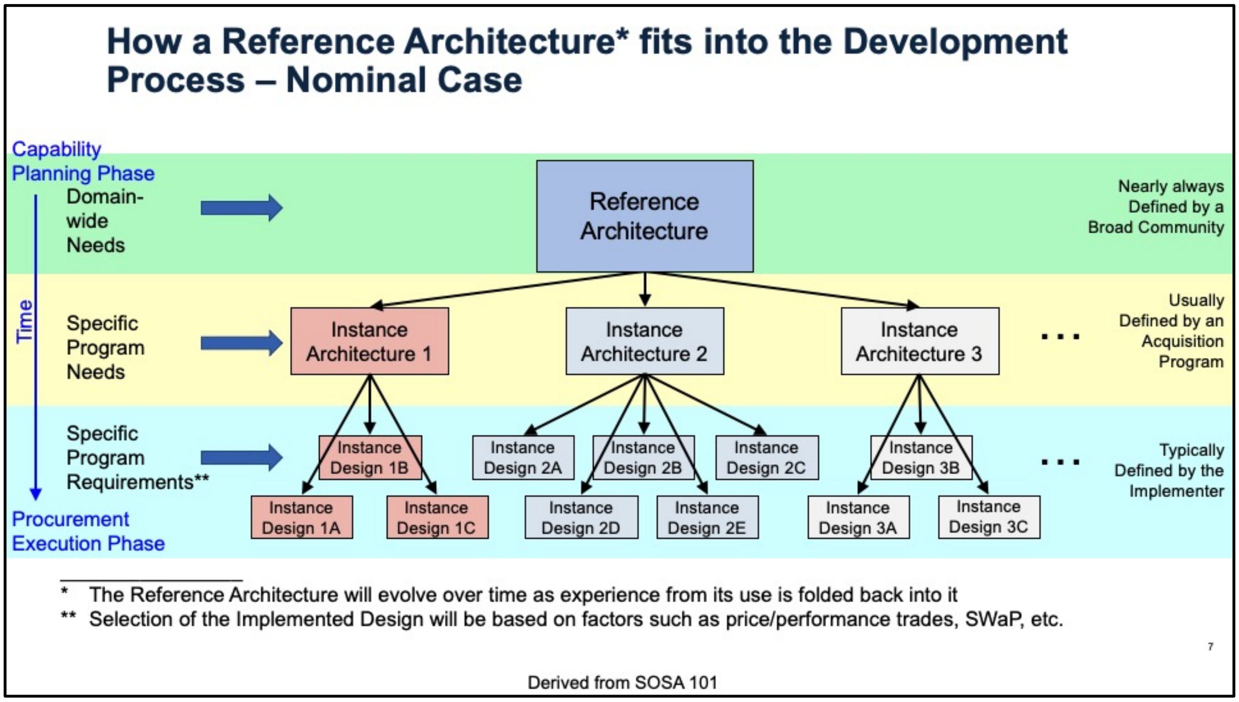
Many people are confused by the terms MOSA and OSA (especially since the "A" means different things). The *approach* has a business and a technical side, and a business side. The business side is focused on acquisition (processes, practices, and policies). The technical side is quite simple: use modularity and leverage open systems architectures and standards.

- It should be pointed out that there are quite a few OSAs out there, each one created to solve a specific problem or address a particular set of needs. Contrary to some myths (and with apologies to Lord of the Rings), there is no "one OSA to rule them all" – each OSA has its place
- Where the business and technical sides meet is how you confirm that the modules and interfaces making up making up a system are in fact using the consensus-based standards and OSAs correctly. That is the role of Conformance Verification.



Many people are confused by, or completely misunderstand, the term "architecture." The word does not mean the "DoDAF-ication" of a design. This definition captures it nicely. It all boils down to modules (which encapsulate functionality and exhibit behaviors) and the interfaces that connect them. Period.

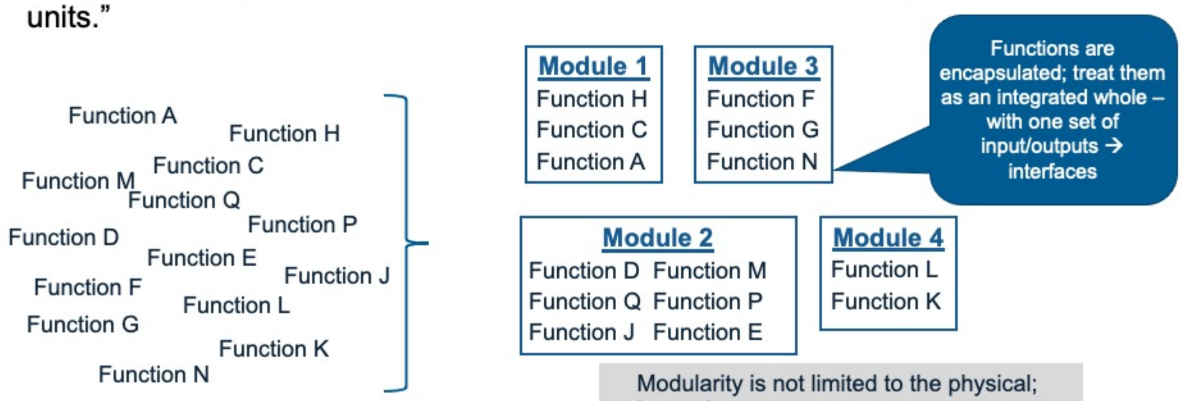
Modules are the building blocks of the architecture, they have interfaces and at those interfaces the exhibit behaviors. That's the essence of artchiteture.



Ideally, the Reference Architecture is used to ensure the Instance (subordinate) architectures all have common lineage, and can achieve the broader vision.

Modular Defined

"… functionally is partitioned into discrete, cohesive, and self-contained units with well-defined interfaces that permit substitution of such units with similar components or products from alternate sources with minimum impact on existing units."



Modularity is not limited to the physical; software functionality can be modularized too

MITRE

Modular Interfaces

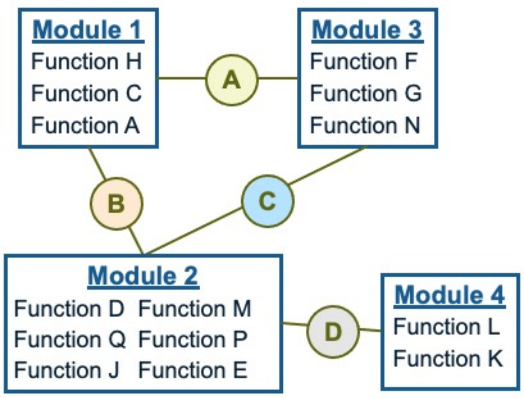
Module 1
Function H
Function C
Function A

Module 3
Function F
Function G
Function N

Module 2
Function D Function M
Function Q Function P
Function J Function E

Module 4
Function L
Function K

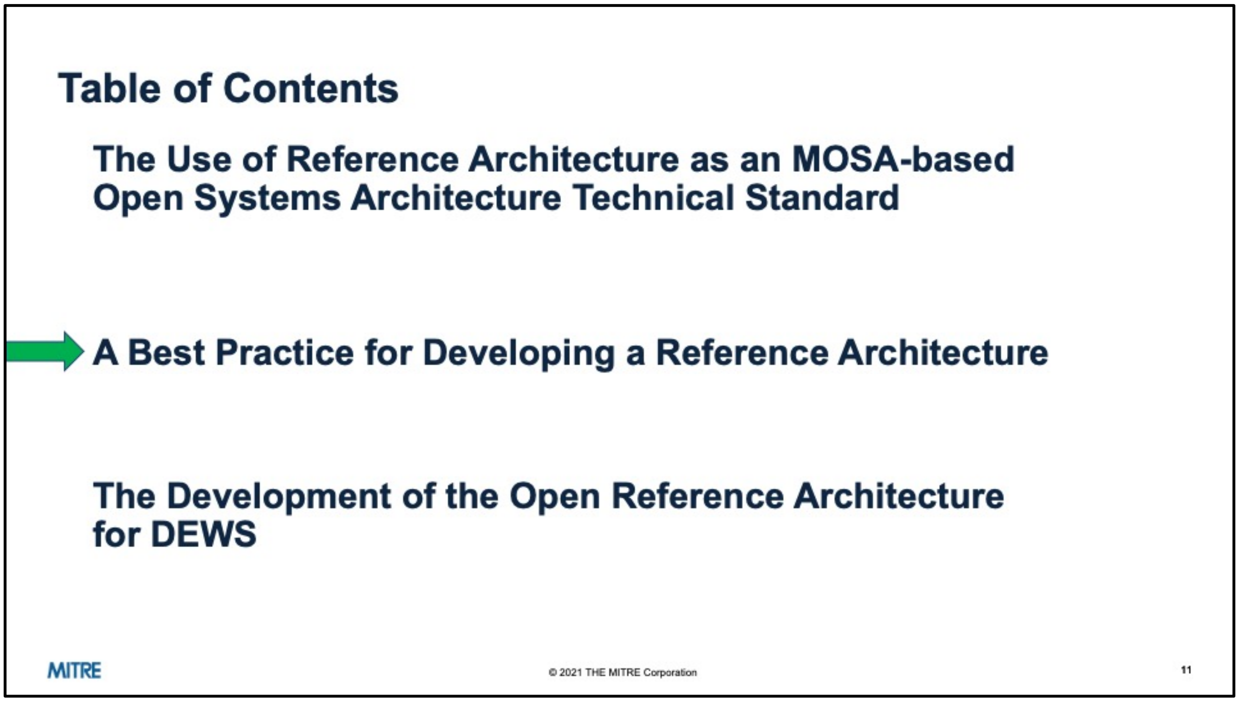
Modular Interfaces



Toy Example of Interactions Documented

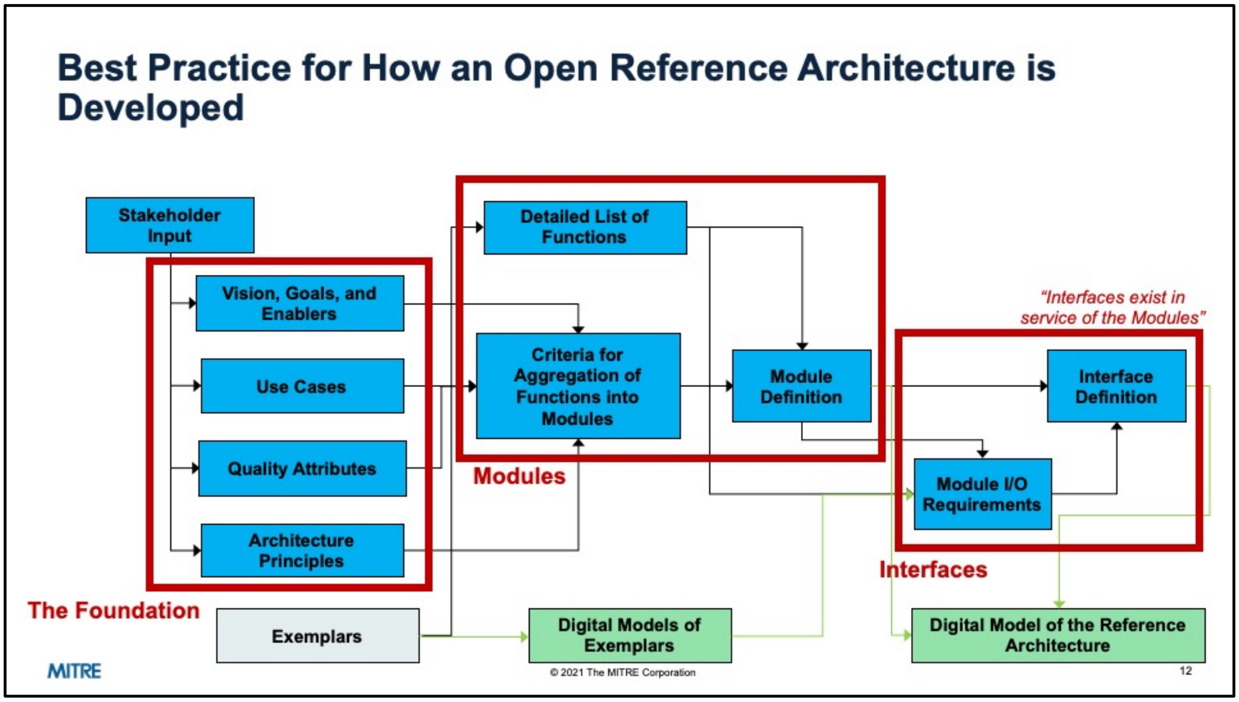
Interaction on Interface	Source Module	Dest. Module	Conveyed Signal&Data
A1	1	3	Rx Signal
A2	3	1	Tasking
A3	3	1	Interlock
B2	2	1	Enable
C1	3	2	Tracks
C3	2	3	Cues
C4	3	2	Signal Quality
C5	2	3	Track Priority
D1	2	4	Write data
D2	4	2	Read data

Interactions between Functions within the same Modules do not appear on Interfaces



In the course of this presentation, I am going to answer these three questions: Why does MOSA Matter? What the heck is MOSA, anyway? And How is MOSA being applied to Directed Energy systems?

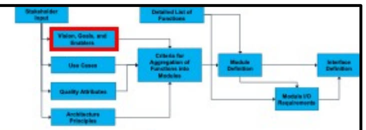
Let's begin with "why..."

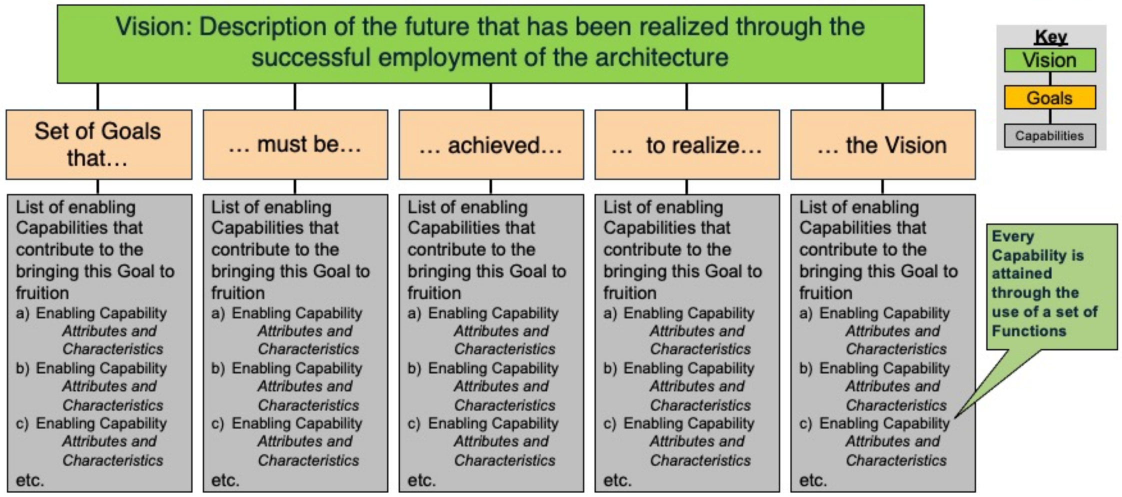


The DEWS MOSA Reference Architecture was developed in four phases: The first established the care-abouts and involved getting stakeholder consensus.
The second defined the Modules that make up the architecture
The third defined the interfaces between the Modules
And in the fourth we incorporated the architecture into a digital model – as the single source of truth

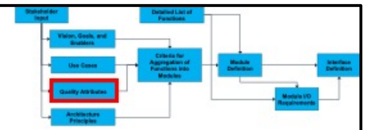
All along the way, we used existing and emerging DEWS systems as exemplars.

Architecture Driven by the Vision, Goals, and Enabling Capabilities





Quality Attributes: How you know your architecture meets the needs of the stakeholder community?



- Means of estimating the "goodness" of an architecture
- Often captured as "ilities" – properties that matter
- Based on stakeholder consensus
- Rank-ordered (so you can make decisions/trades)
- Typically arrived at during a Quality Attributes Workshop (QAW) at the beginning of architecture development (so everyone is on the same page)
- Don't let the group pick more than 7-10 Quality Attributes – you can't have it all

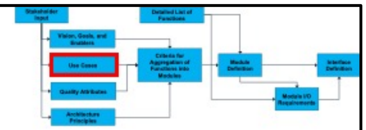
MITRE

Qualities of Architectural Principles

- Understandable: Can be quickly grasped and understood by individuals throughout the organization -- clear and unambiguous, so that violations (whether intentional or not) are minimized
- Robust: enable good quality decisions about architectures, and be sufficiently definitive and precise to support consistent decision-making in complex, potentially controversial situations
- Complete: Cover every aspect and anticipated situation
- Consistent: Non-contradictory with others; adhering to one principle would violate the spirit of another – so every word in a should be carefully chosen
- Stable: Enduring, yet able to accommodate changes (through an amendment process)

MITRE

Use Cases

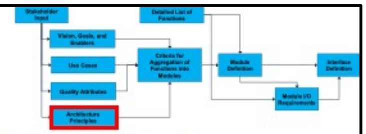


- Use Cases capture how the systems that result from the architecture will be used
  - Based on stakeholder input
  - Both near-term and (even more importantly) long-term
  - Generalized to ensure broad applicability (not "CG-62 moves north along the coast, and at 0350 hrs a target bearing 162 deg approaches")
- The Use Cases must fully characterize actions, functions, interactions, and behaviors
- The resulting OV-1s, OV-2s, OV-5bs (yes, plural!) become the foundation for the architecture

If you don't have a solid understanding of these, you run the very serious risk of developing an architecture that doesn't do what is needed

MITRE

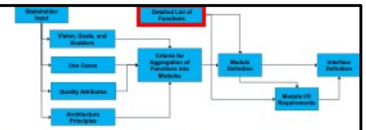
Architecture Principles



- Architecture Principles are general rules and guidelines that guide/inform and support the development and maturation of an architecture
- They are to be enduring and seldom (if ever) amended
- They form a basis for decision-making
- They govern the architecture process, affecting the development, maintenance, and use of the architecture
- They typically contain four parts:
  - Short name (nickname)
  - Statement of what it is
  - Rationale (why it matters)
  - Implications for the architecture

MITRE

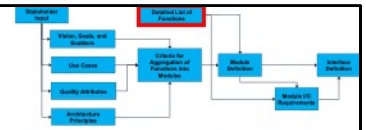
Function Assessment



- A reference architecture is a superset that contains all the functionality that could possibly be instantiated
  - Particular instantiations may or may not have all the functions – but this is a Reference Architecture so account for them all

- Document all the functions performed by a hypothetical superset system
- For each, identify the Inputs and Outputs (including control)

Module Definition Process

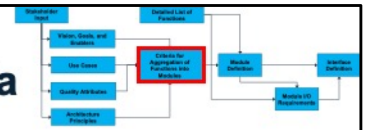


- Compile a detailed list of the functions that are performed within a superset system
- Generate a rough aggregation of functions into modules
- Analyze how different aggregations stack up against:
  - Achieving the Quality Attributes as “tested” by the Use Cases
    - Priority given to Use Cases that are ranked higher
  - Consistency with Architecture Principles
  - Supporting the Vision, Goals, and required Capabilities (CV-1, CV-3)

The product of this modularization process is the foundation for the interface definition

MITRE

Module Definition Process: Typical Criteria



MOSA based on:

- **Tight coupling of functions within modules**
- **Loose coupling between modules**

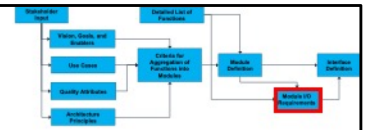
Example Considerations and Resultant Actions:

Consideration	Resultant Action
Protection of IP	Ensure functions whose interaction would expose IP are within the same module
Long Operational Life	Ensure isolation of functions that may quickly become obsolete
Module Portability	More encapsulated that can be separated and used elsewhere
Rapid Upgrade	Include enough functionality that modules are independently testable
MOSA Compliance	Tight coupling of functions within modules, loose coupling between modules, standards-based interfaces
Cohesion	Combine functions with high cohesion to minimize complexity of interfaces or have that have common I/O needs
Variants	Function aggregation must support resulting Modules and Interfaces that have “small / medium / large” variants

This process is iterative, multiple options weighed, decision tree-like

MITRE

Modules → Interfaces



Interfaces exist in service of the modules (not the other way around)

For each function within a Module, identify

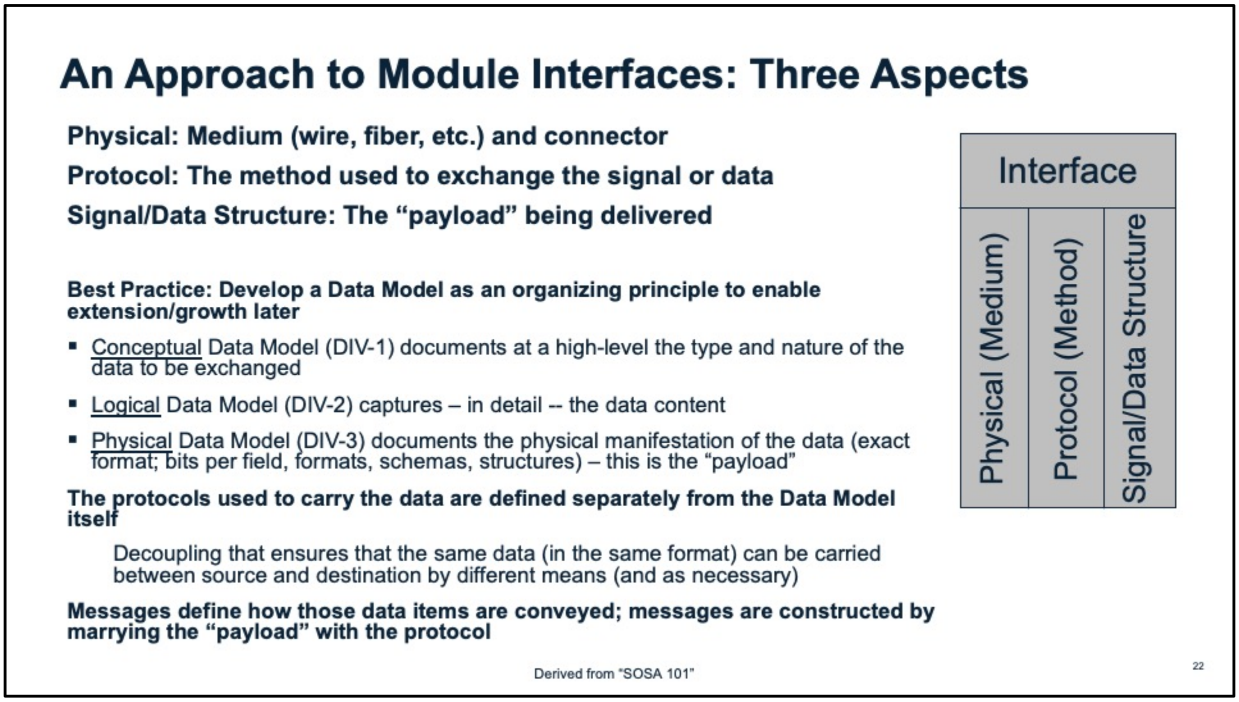
- What is required for input (not provided by another function inside that Module)
- What it produced for an output that is used outside the Module

Generate module I/O requirements

- For each required input, identify the source module
- For each produced output, identify the target module(s)

Interfaces are where you assess conformance (or compliance) to the architecture

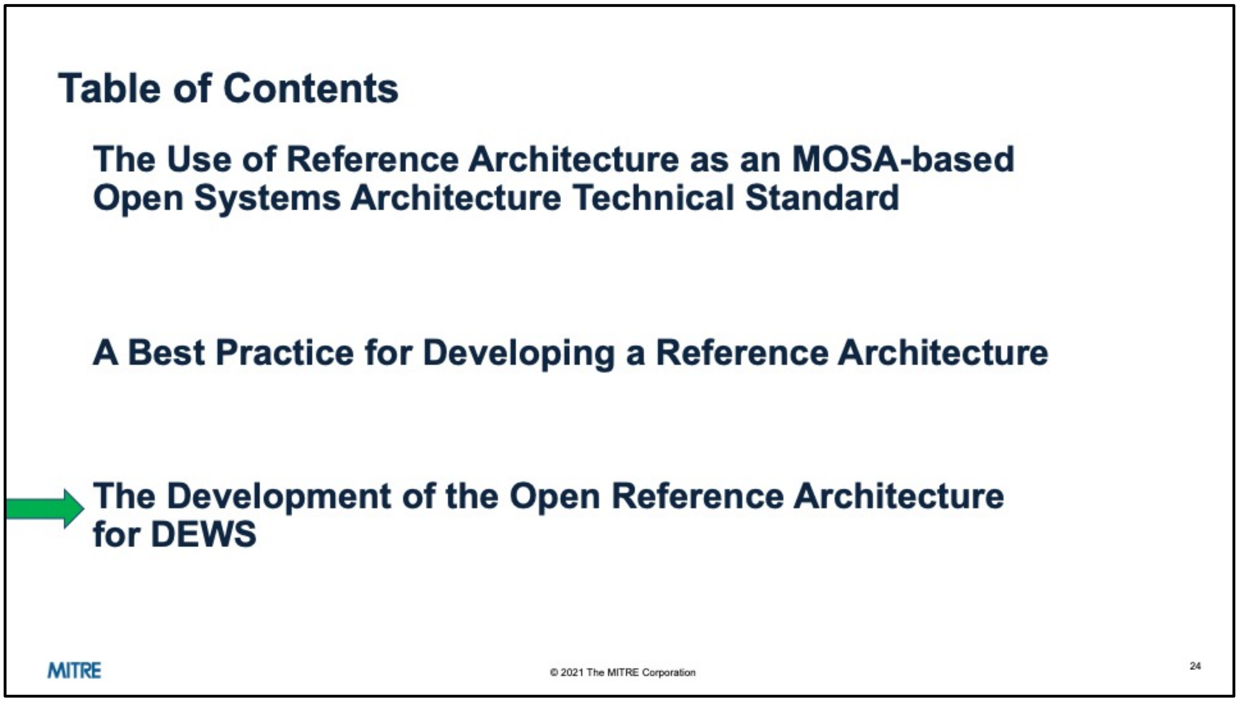
Derived from "SOSA 101"



One can think of interfaces as having (at least) three dimensions: Physical (includes media/cabling/fiber), the Protocol (how signal or data flows, such as framing or handshake), and the Data or Signal structure or encoding itself. Our approach is to take a top-down view (not generate a pile of messages) by developing a Data Model that creates the organizing principle around the data, and then define the means of conveyance.

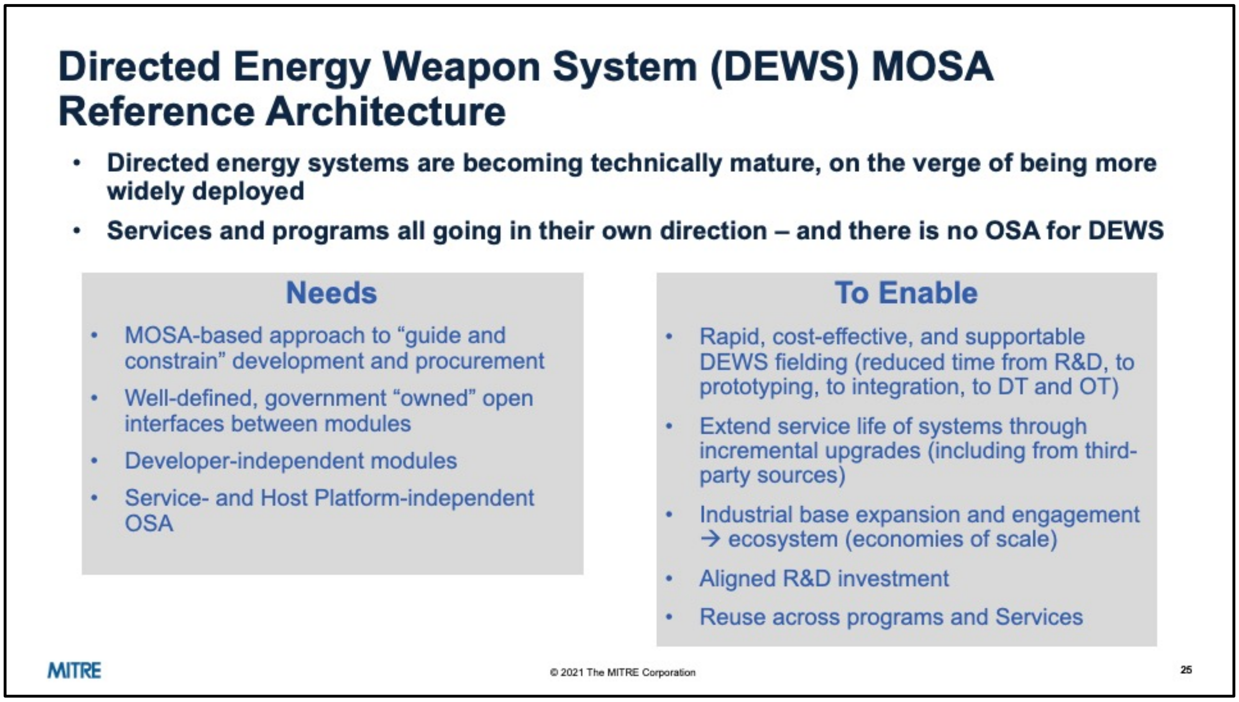
A fully-formed Data Model has three levels of definition. The top-level is called the Conceptual Data Model (or DIV-1 in DoDAF parlance) and defines the nature and relationship between data entities. The Logical Data Model (DIV-2) depicts the data content in terms of units and bounds – and is a higher degree of specificity (e.g., for a "Track" defined in the DIV-1, the associated DIV-2 entities would include latitude as 0 to 360deg, Longitude as -90 to +90deg, and altitude as meters above mean sea level). The Physical Data Model (DIV-3) specifies the physical representation (e.g., floating point, double precision, signed integer, etc.) of the DIV-2 entitles.

We separate the protocol from the data model in order to maintain flexibility. For example, if I am giving you a phone number, I might just give you all ten digits and you would then nod that you got it. That's fast, but prone to error. I might give you the area code, and you nod, then the prefix, and you nod, and then the last four, and you nod. That's only slightly more reliable but you can say, 'Repeat that last part.'. I might give you all ten digits, and you read them all back to me. Of course, if you missed something then I would have to make a specific correction or read it all again. I might give them to you in chunks as before and you repeat each chunk back to me – and if there was an error, we would catch it quicker (this is the most reliable, albeit the most clumsy and slow). The point is that these are different protocols for conveying <u>the same data</u>, each has its benefits and limitations. And what we say, and how we say it is independent of the pathway over which it is conveyed (tin cans and string, cell phone, shout across the room, etc.) – and who and I are. These are separate and independent aspects of the interface."



In the course of this presentation, I am going to answer these three questions: Why does MOSA Matter? What the heck is MOSA, anyway? And How is MOSA being applied to Directed Energy systems?

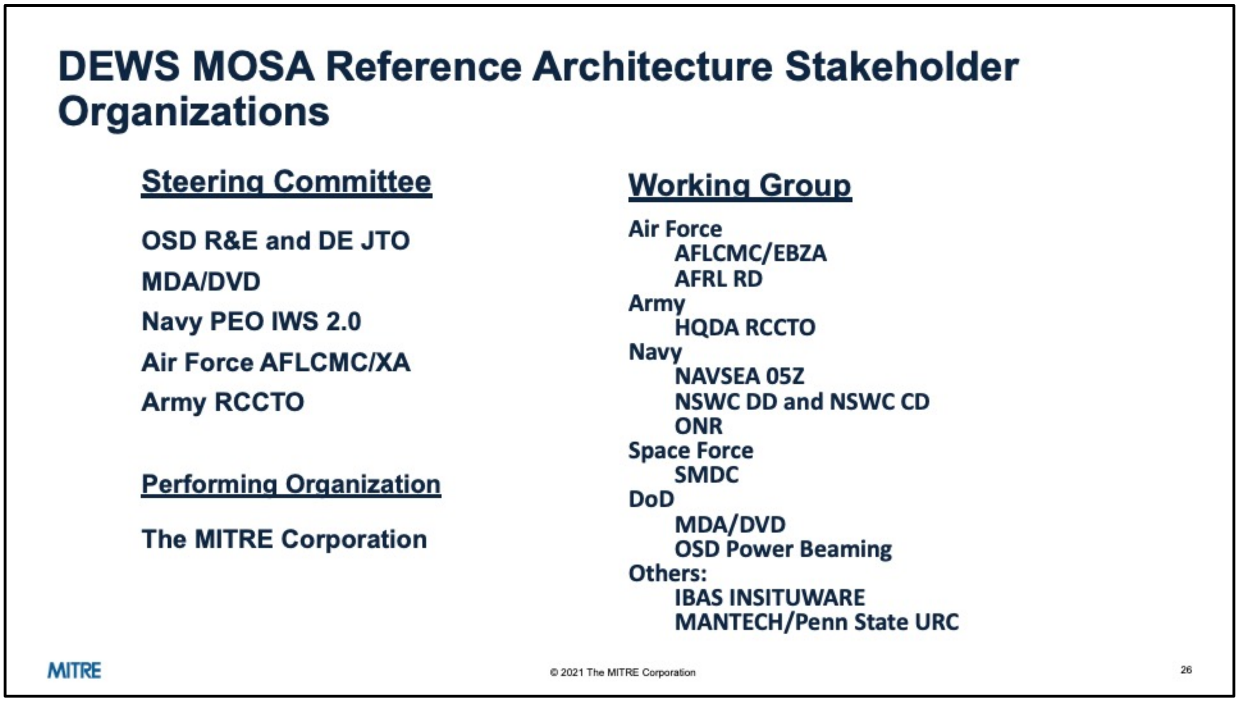
Let's begin with "why..."



We are all here because Directed Energy systems are maturing to the point of being fielded

I mentioned before that there are many OSAs. But there isn't an OSA in existence that addresses directed energy systems – until now. Prior to this work, there is no equivalent of OMS, FACE, SOSA, or VICTORY for DEWS – nothing directly applicable to Directed Energy systems.

Now is the time to leverage a DEWS open reference architecture so the DoD – as an enterprise – can realize benefits of economies of scale, a healthy industrial base, and maximize reuse and make the most of investment dollars



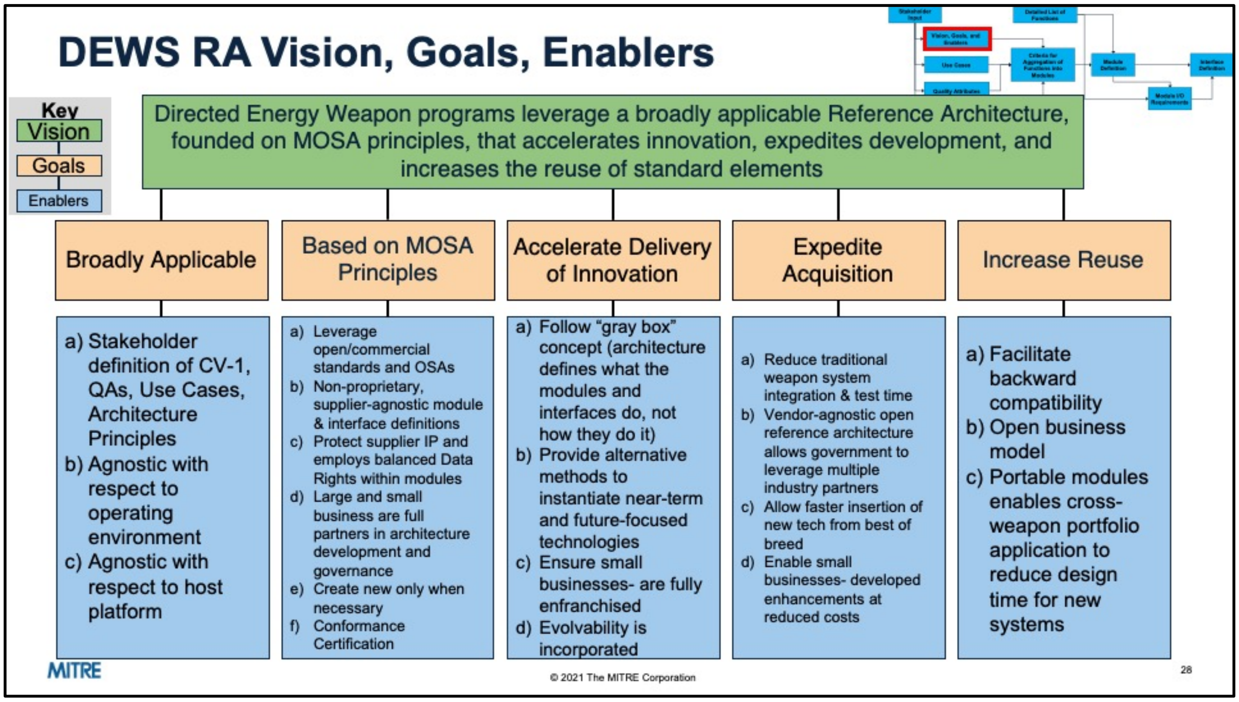
The DEWS MOSA Reference Architecture is being developed by MITRE under the oversight of a very Joint Steering Committee and Working Group that guides and advises us.

Exemplars: Learn from Systems under Development

Program Name	Service	Type	Domain	Government Lead	Industry Partners
Bane	Navy	HPM	Land	Dahlgren DEW Office (DEWO)	
Tactical High-Power Microwave Operational Responder (THOR)	USAF	HPM	Land	AFRL	BAE Systems (BAE), Leidos, Verus Research
DE-Maneuver Short Range Air Defense (MSHORAD)	Army	HEL	Land	Army RCCTO	Kord Technologies (Kord)
Indirect Fire Protection High Energy Laser (IFPC-HEL)	Army	HEL	Land	RCCTO, OSD, SMDC	Dynetics (a Leidos Co.), Lockheed Martin Corporation (LMC), MZA Associates Corporation (MZA)
High Power Joint Electromagnetic Non-Kinetic Strike (HiJENKS)	Navy: ONR USAF: AFRL	HPM	Air	Dahlgren /Kirtland	LMC
Self-protect High Energy Laser Demonstrator (SHiELD)	USAF	HEL	Air	AFRL	LMC
Airborne High Energy Laser (AHEL)	USAF	HEL	Air	AF Special Operations Command (AFSOC)	
High Energy Laser with Integrated Optical-dazzler and Surveillance (HELIOS)	Navy	HEL	Sea	PEO IWS 2	LMC
Layered Laser Defense (LLD)	Navy	HEL	Sea	ONR	LMC
Solid State Laser Technology Maturation (SSL-TM)	Navy	HEL	Sea	Dahlgren	Northrop Grumman

MITRE

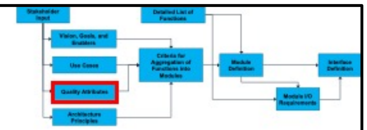
This list shows the exemplars we used. Note that we drew from a wide variety of systems, laser as well as microwave, from across the Services



The first thing we did was to get across-the-board agreement on the Vision, Goals, and Enablers.

Through a lot of interaction, the Joint community agreed on what you see on the screen now. Time doesn't permit me to go into details, but you see that we established a common Vision, which will be attained by meeting a set of Goals, and these Goals are achieved by realizing a set of Enabling Conditions -- which are ultimately created by the existence of the DEWS Reference Architecture.

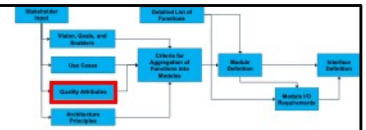
DEWS Open Reference Architecture Quality Attributes (1 of 3)



#	Quality Attribute	Description
1	Modularity	The property of a system whereby it is composed of individually distinct physical and functional units that, while possibly tightly coupled internally, are loosely coupled with one another. In the context of the DEWS MOSA RA, this quality attribute enforces the establishment of fully-defined standardized system modules (encapsulated functionality, behaviors, and interfaces) that can be individually developed and verified for conformance to the published DEWS MOSA RA.
2	Upgradability	The ability of a system to be improved or evolved without the need to make significant architectural changes. In the context of the DEWS MOSA RA, this quality attribute means that the architecture was created with the explicit intent of supporting upgrades in terms of technology (so that rapid change is deliberately encapsulated inside specific modules) and performance (modules and interfaces definitions incorporate options for good/better/best).
3	Portability	An attribute that describes the reuse of elements in other systems without the need to create new (or adapt old) ones for each system. In the context of the DEWS MOSA RA, this quality attribute refers to the ability of DEWS Modules to be used, with minimal (or ideally no modification), in other DEWS MOSA RA-based environments (e.g., different operational domains, different systems, similar environmental requirements, etc.).

MITRE

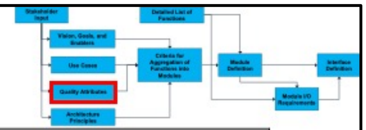
DEWS Open Reference Architecture Quality Attributes (2 of 3)



#	Quality Attribute	Description
4	Scalability	The ability of a system to be adapted to or integrated with higher levels of architectures; the capability of an architecture to support designs that can be adapted to increased or expanding performance demands without requiring architectural changes. In the context of the DEWS MOA RA this Quality Attribute means that more capable versions of a resultant DEWS can be created by either combining multiples of like modules or replacing modules with higher performing equivalents (and possessing the same interfaces) in order to build up the necessary capability.
5	Interoperability	The ability of a system to exchange data with other systems, and for both to use the exchanged data to operate (even more) effectively together. In the context of the DEWS MOSA RA, this quality attribute refers to the ability of DEWS to be able to exchange data and Command and Control (C2) information with other systems (e.g., other combat systems, host platforms, and non-kinetic and kinetic weapons), possibly through adaptation if needed.
6	Safety	The ability of a system to operate without injury to the operator, crew, nearby innocents – and without harm to the system, host platform, and unintended targets. In the context of the DEWS MOSA RA, this quality attribute means that the architecture includes features to protect personnel and property (including satellites), and that nothing in the architecture will preclude a resultant system from passing safety accreditation (NB: Safety accreditation is evaluated awarded on the instantiated design, not Reference Architecture).

MITRE

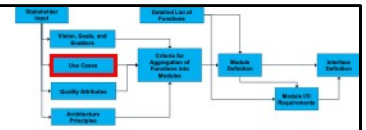
DEWS Open Reference Architecture Quality Attributes (3 of 3)



#	Quality Attribute	Description
7	Reliability	The property of a system to timely and dependably perform its intended function in a manner that makes it operationally effective. In the context of the DEWS MOSA RA, this quality attribute provides that measures will be incorporated into the architecture to minimize the likelihood that a system based on the architecture will fail to perform as designed.
8	Configurability	A property that enables the user of a system to set up the hardware and/or software in different ways in order to meet immediate operational needs. In the context of the DEWS MOSA RA, Configurability means that users of the systems that are derived from the architecture will have the option to make some in-the-field alterations (typically software or firmware changes vetted through security accreditation, or physical dismounting of architecturally-aligned DEWS Modules) in order to adjust performance and behavior to meet dynamic mission needs.
9	Resiliency	The ability of a system to respond to disruption or over-capacity (either naturally occurring or man-made, inadvertent or deliberate) and continue to operate (or return to normal operations), perhaps with graceful degradation of performance. In the context of the DEWS MOSA RA, this quality attribute refers to the ability of DEWS to be able to maintain normal (or near-normal) operation while impacted by electronic interference, cybersecurity attack, or limited physical damage.
10	Deployability	The ability of the system to be installed and easily fielded on various host platforms and in various environments. In the context of the DEWS MOSA RA, this quality attribute means that designs resulting from the DEWS MOSA RA will have host platform interfaces that will make it a relatively simple matter to install /de-install a DEWS on/from the host platform or to move it to other host platforms – enabled through a Host Platform Module.

MITRE

Use Cases

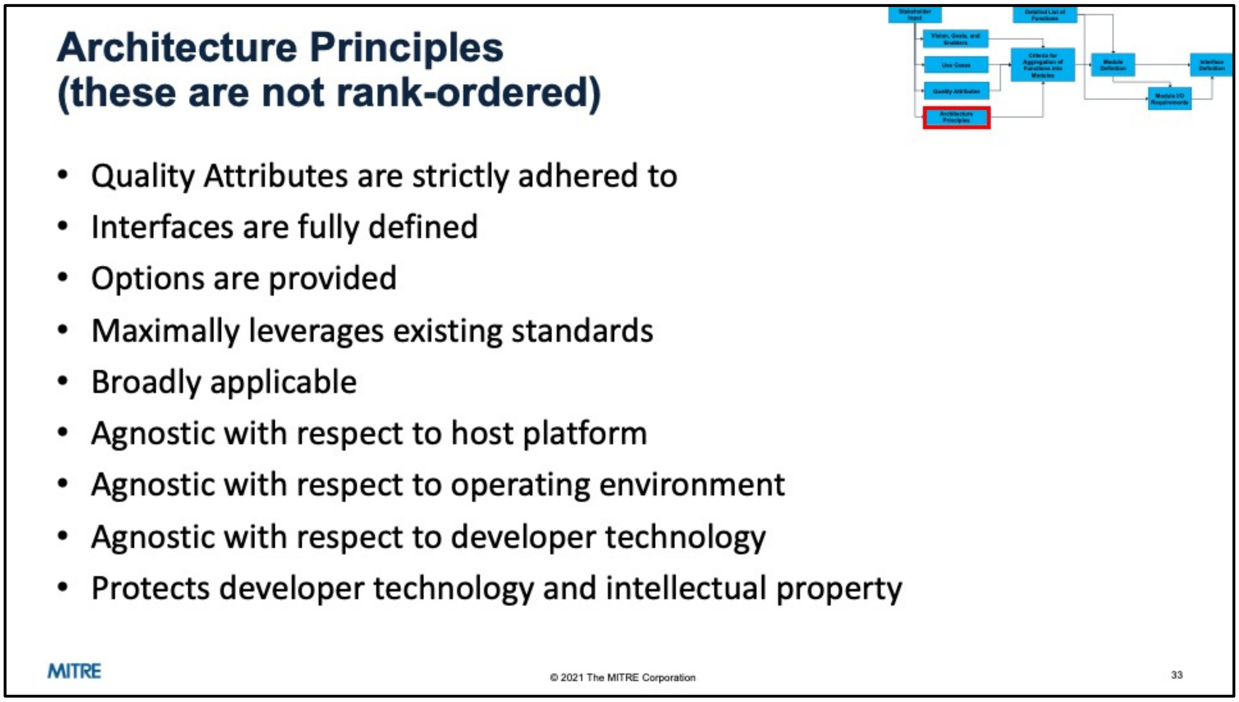


Surface / Air C-UAS
Ground / Air C-UAS
Surface / Air ASCM
Hypersonic Defense

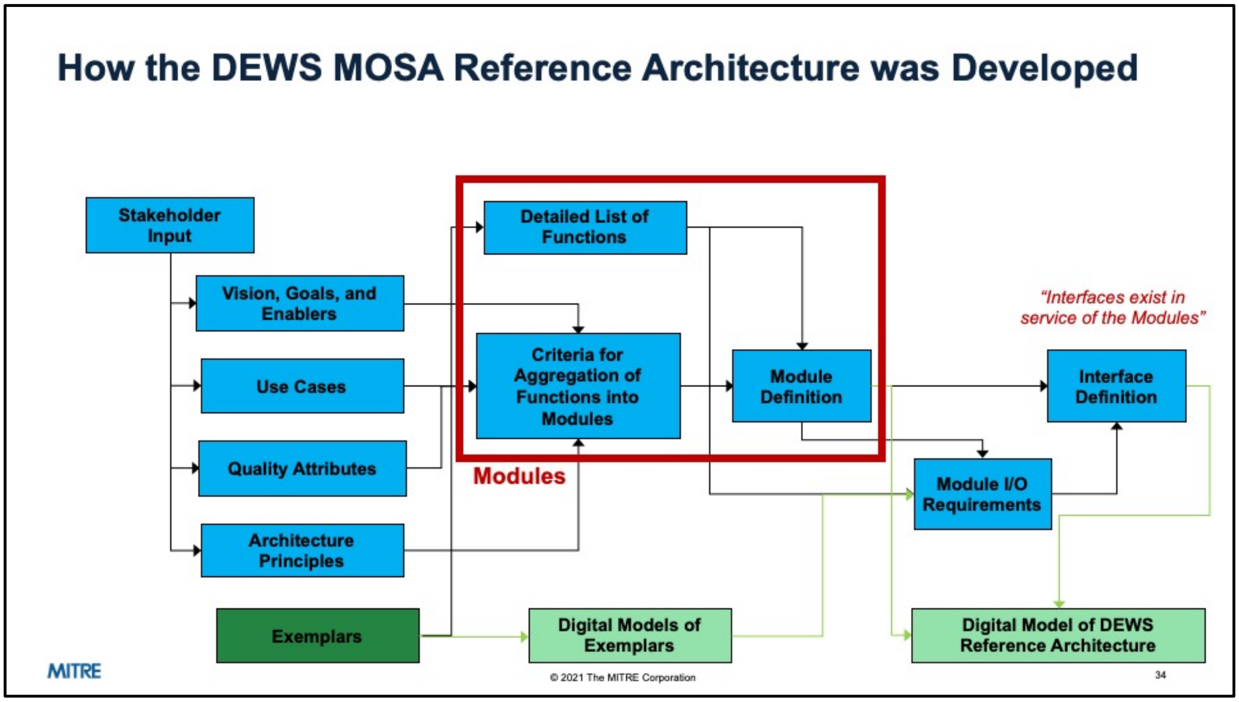
Ground C-IED
Air / Air Combat
Air / Ground HEL
Air / Ground HPM
… among others

MITRE 32

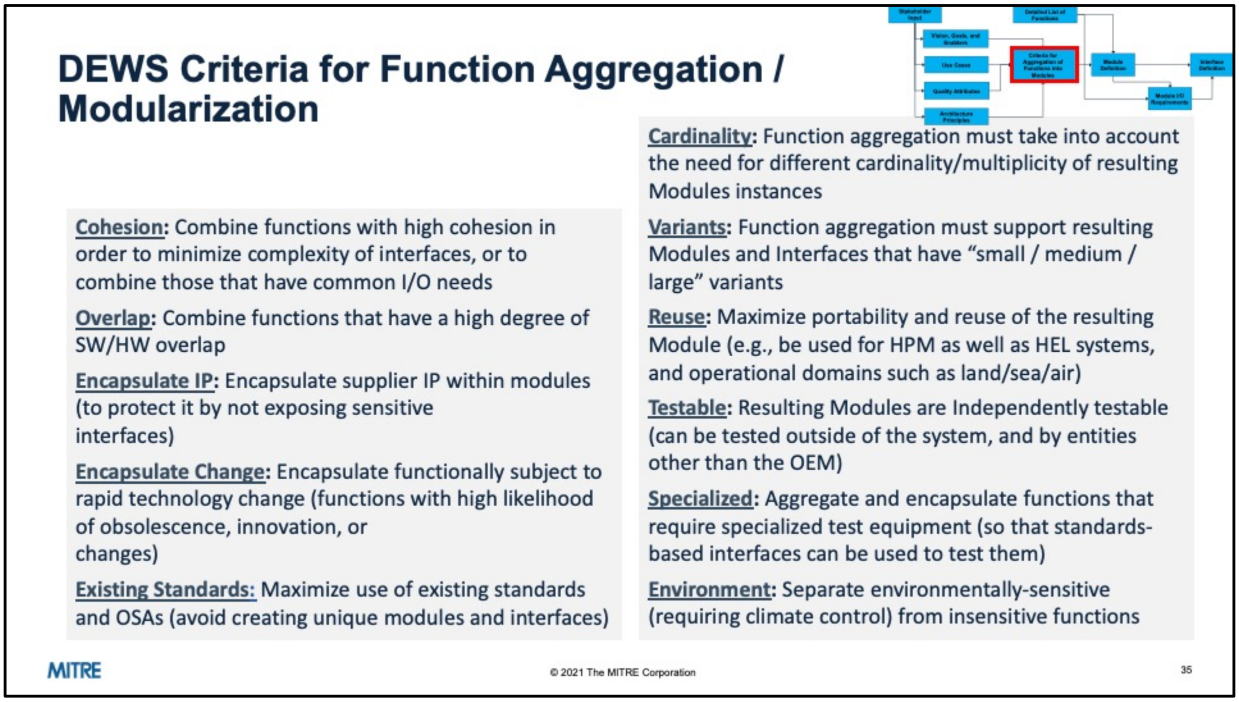
And finally, we have a set of Use Cases that capture how a resultant DEWS system will be used



And we have a set of “run rules” for the development of the DEWS MOSA Reference Architecture were also developed and fully vetted. For example, in order to be broadly applicable, the DEWS MOSA RA has to be agnostic with respect to host platform, operating environment, and developer technology.

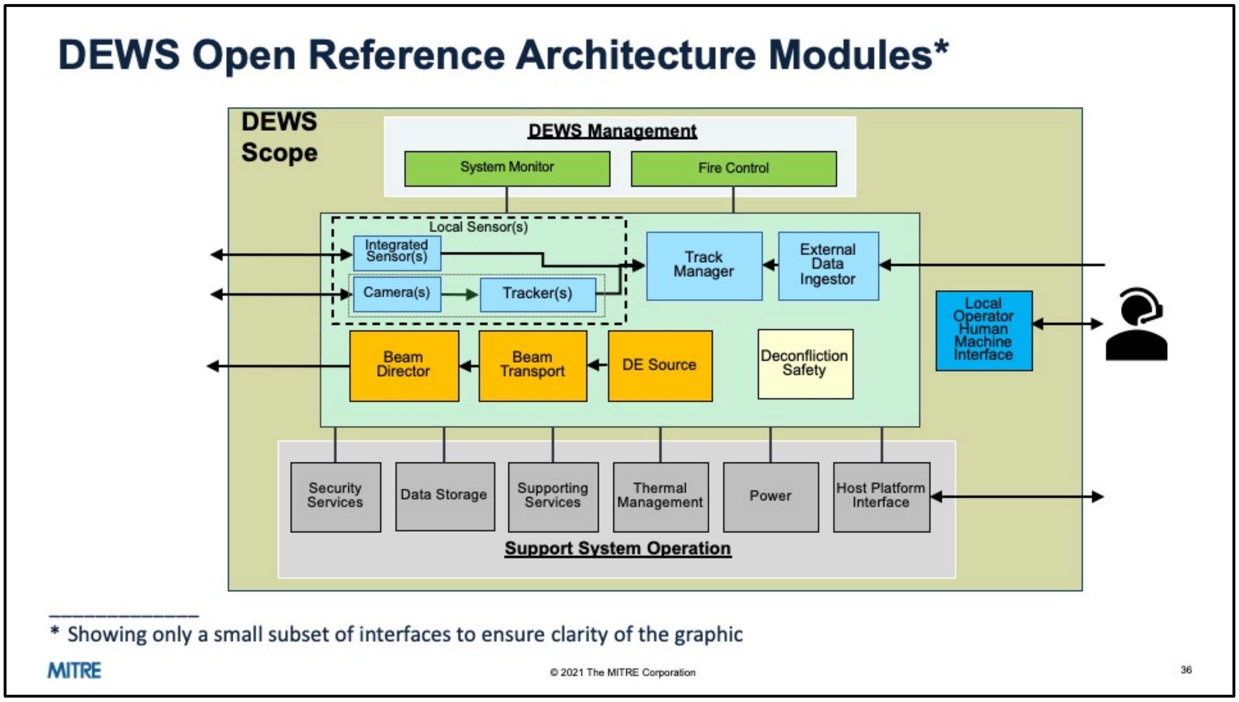


Once we defined the care-abouts we focused on Module development. Here we started with a set of criteria for how the the several hundred functions that make up a DEWS system would be aggregated into the Modular entities



The criterial included “Cohesion” -- consistent with MOSA principles – to combine functions that are tightly coupled into modules, ensuring that there is loose coupling between modules.

In addition, and conscious of the need to encourage industry innovation, the modules were defined so that intellectual property would be contained inside modules and not exposed across open inter-module boundaries.



And here they are: The Modules that make up the DEWS MOSA Reference Architecture. Two (shaded green) have to do with management, six (shaded gray) provide services to the rest, a cluster (shown in blue) are associated with object detection and tracking, and three (in orange) are engaged with delivering the energy to the target. Plus, we have one each for deconfliction/safety, and one as the human-machine interface for an optional local operator.

Note that only of the few interfaces are shown here, for illustration purposes only. If we included them all, the chart would be unreadable.

Example of Functions Encapsulated in one Module

ID	Name	Description
21.11	Display Situational Awareness Data	Display map of system tracks relative to host platform. May include symbology (e.g., MIL-STD-2525), overlays, etc. Display sources include result of the current track store, Local Sensors, primary aperture, or direct feed from External Sensor or Host Platform. This display is also a user interface for functions such as designating tracks. If a track is to be designated for attack, the designated object is tagged as the target
21.13	Display Video	Display Local Sensor video feeds to DEWS operator for use in carrying out engagements. The feeds may be real-time or pre-recorded. The operator display allows the replay, pause, rewind, fast forward, etc. (so-called "TiVo functionality") permitting reconstruction and analysis.
21.14	H&S Status Update	Request and receive system H&S from the System Monitor
21.15	Display and Control Status	Provide visual display of system status (received from Fire Control), view and control states/modes (conveyed to Fire Control), system power on/off, fault conditions and alarms (including the ability to drill-down to gather more detail, and clearing alarms)
21.16	Initiate BIT	Request that the System Monitor Module that a Built-in Test (BIT) to be performed

Only showing five of the 26 functions for Module 21 (Local Operator HMI)

MITRE

37

- Digging down inside a Module you will find a few dozen encapsulated functions
- This is a snippet of what's inside one of the modules, where we've articulated a numbering scheme, each function has a concise name, and an extended description.
- The interfaces are derived from the input needs and output products of each of the functions.

Mapping Functions → I/O Needs → Inter-Module Interactions

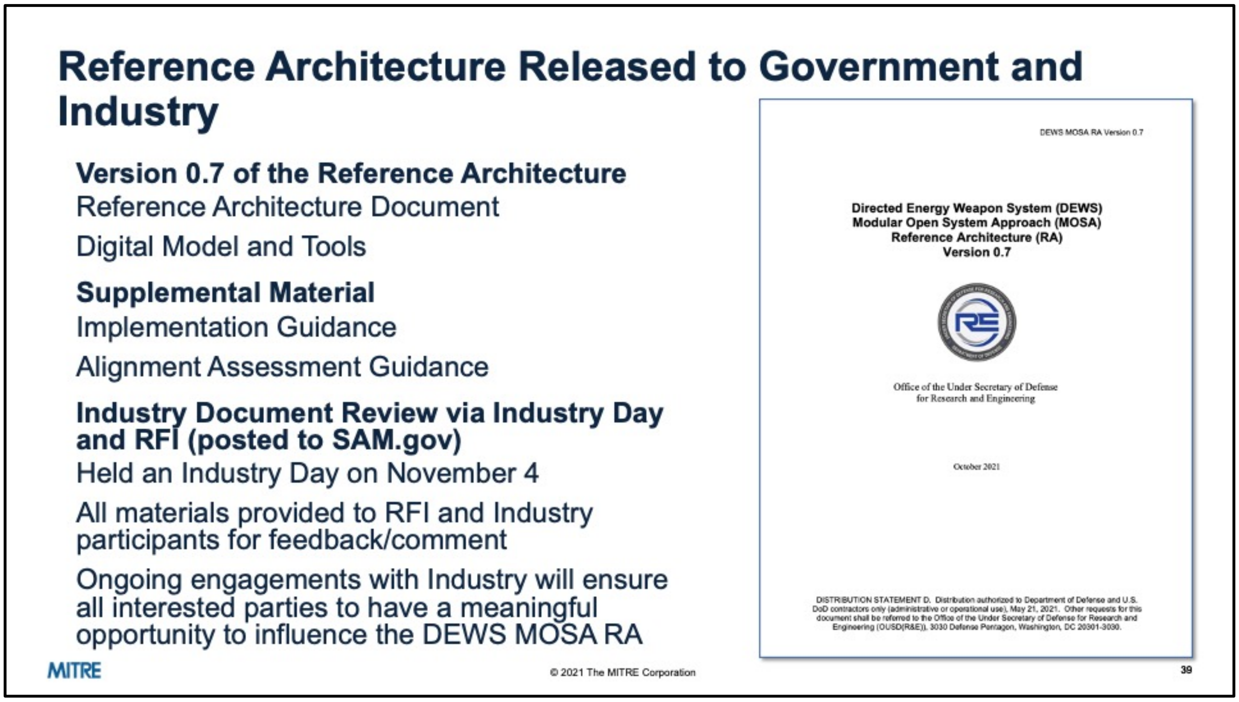
ID	Name	Input Needs	Input Source	Product Produced	Product Destination
21.11	Display Situational Awareness Data	Track Data (tracks and kinematic data)	Track Manager Module	Data in display format	Fire Control Module
21.13	Display Video	Video data	Local Sensor Module (real-time) and Data Storage Module (playback)	Video in display format	(local operator display)
21.14	H&S Status Update	H&S status report	System Monitor Module	Status request	System Monitor Module
21.15	Display and Control Status	System status data,	Fire Control Module	Data in display format, Control messages	(local operator display), Fire Control Module
21.16	Initiate BIT	Operator input	(controls internal to this module)	Request to initiate BIT	System Monitor Module

Only showing five of the 26 functions for Module 21 (Local Operator HMI)

MITRE

38

- Looking at each of the functions within each module, we see their input needs and output products.
- Any time an input comes from a function contained in another Module, we have an interaction. Ditto for products going outside the Module.
- Those interactions aggregate up to form the interfaces between the various, Modules and/or entities outside the architecture (such as the host platform)
- Each of these interactions contain behaviors that were documented as well [use the telephone number example]



All of this, and more, has been released via a SAM.gov announcement as part of an RFI for comment, and we will be holding an Industry Day next week

Our intent is to engage with industry that this becomes a consensus-driven architecture.

Summary

MOSA is effectively implemented using an open Reference Architecture

A best practice was presented, based on both MOSA and core Systems Architecting principles

1. Ensure you understand and document objectives and desired outcomes of the stakeholder community
2. Use these to define the modules ("building blocks") of the architecture
3. Create interfaces that serve the needs of the Modules

An OSA has been developed for Directed Energy systems and is undergoing industry evaluation and feedback

MITRE

[Summarize….]
I'm going to leave this on the screen while I take your questions

Steve Davidson

sdavidson@mitre.org

https://linkedin.com/in/stevedavidson

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™