Co - Evolution of Aeronautical Complex System & Complex System Engineering

Dr. Xinguo ZHANG

Aviation Industry Corporation of China (AVIC)

31 Aug, 2015

Evolution of Aeronautic Complex System

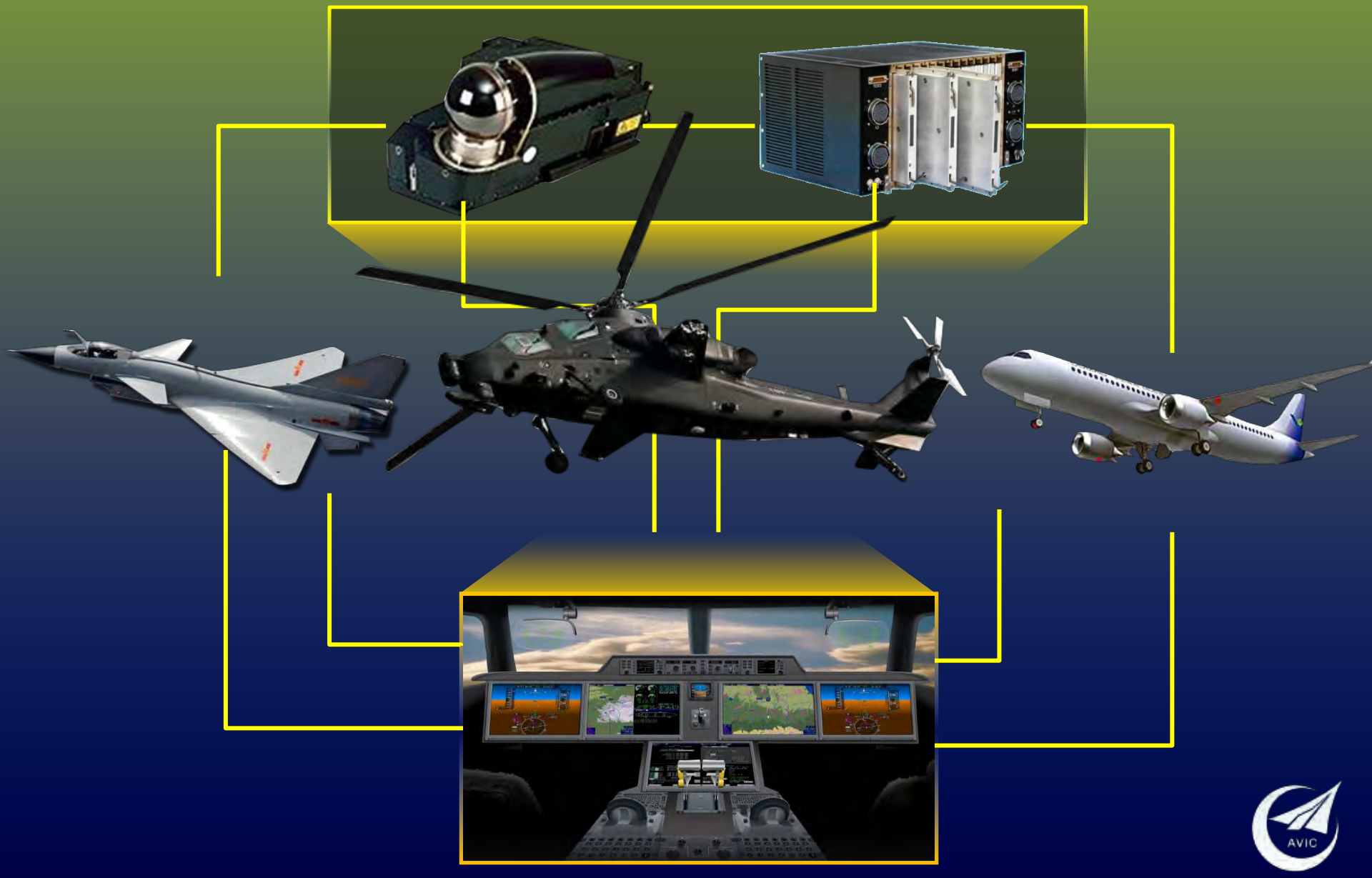
Challenge to Traditional System Engineering

Evolution of Complex System Engineering

Early Product (Mechanical)

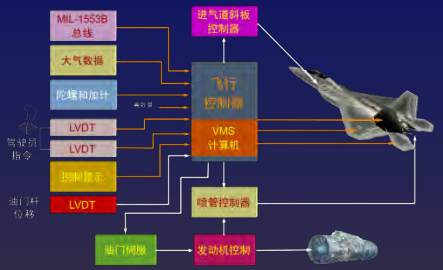
Smart Product (Mechatronic)

Smart Connected Product (Mechatronic/Software/Network)



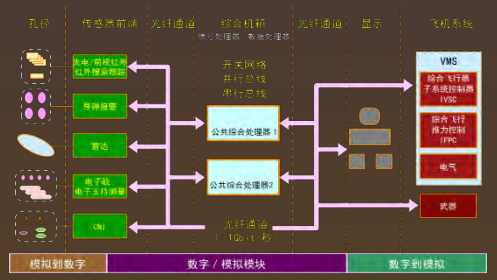
Product System

Flight Control System

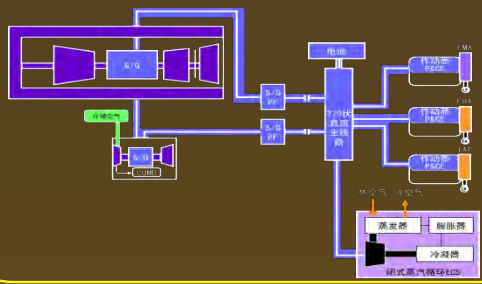


Aircraft System

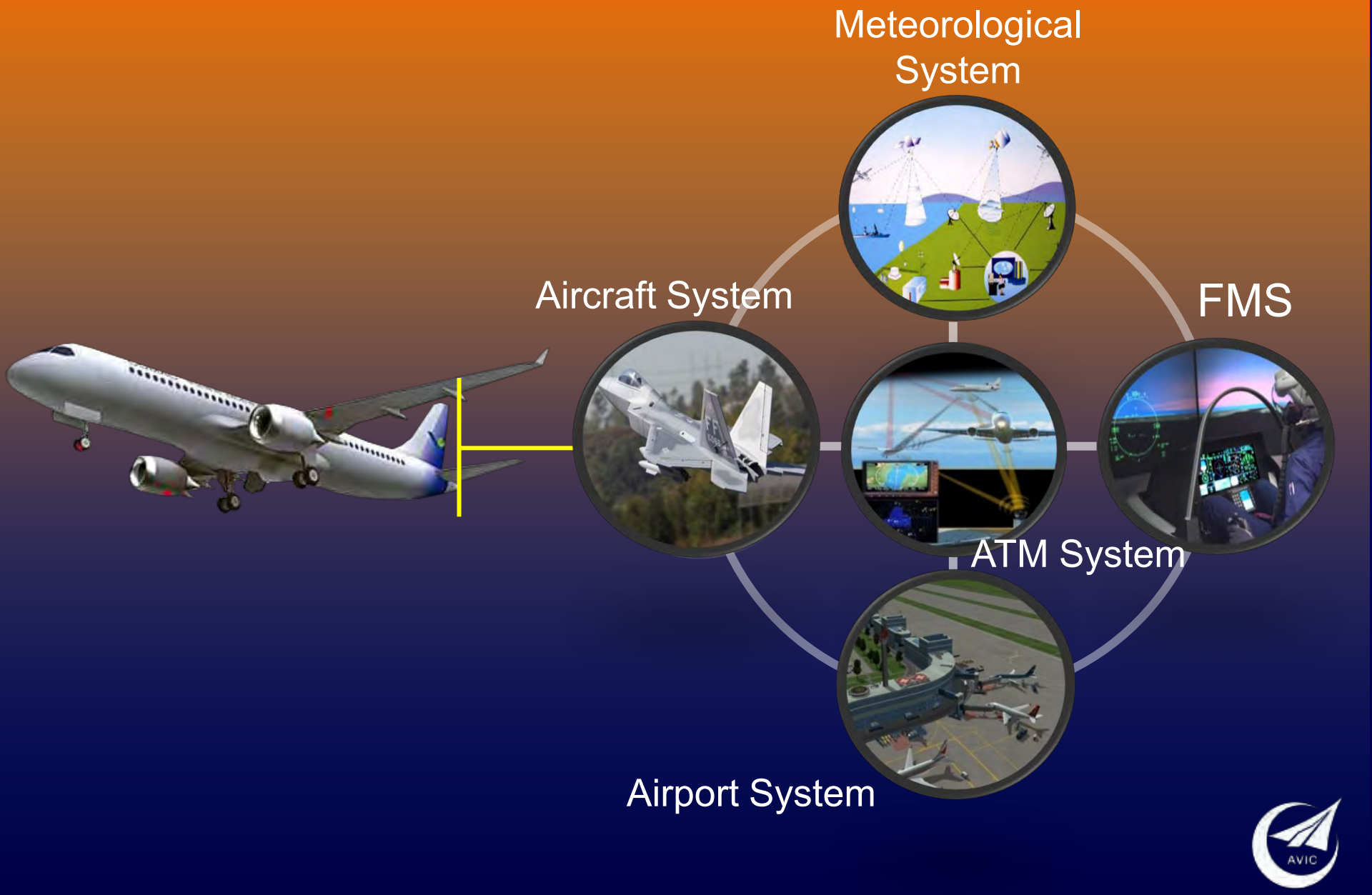
Avionics System



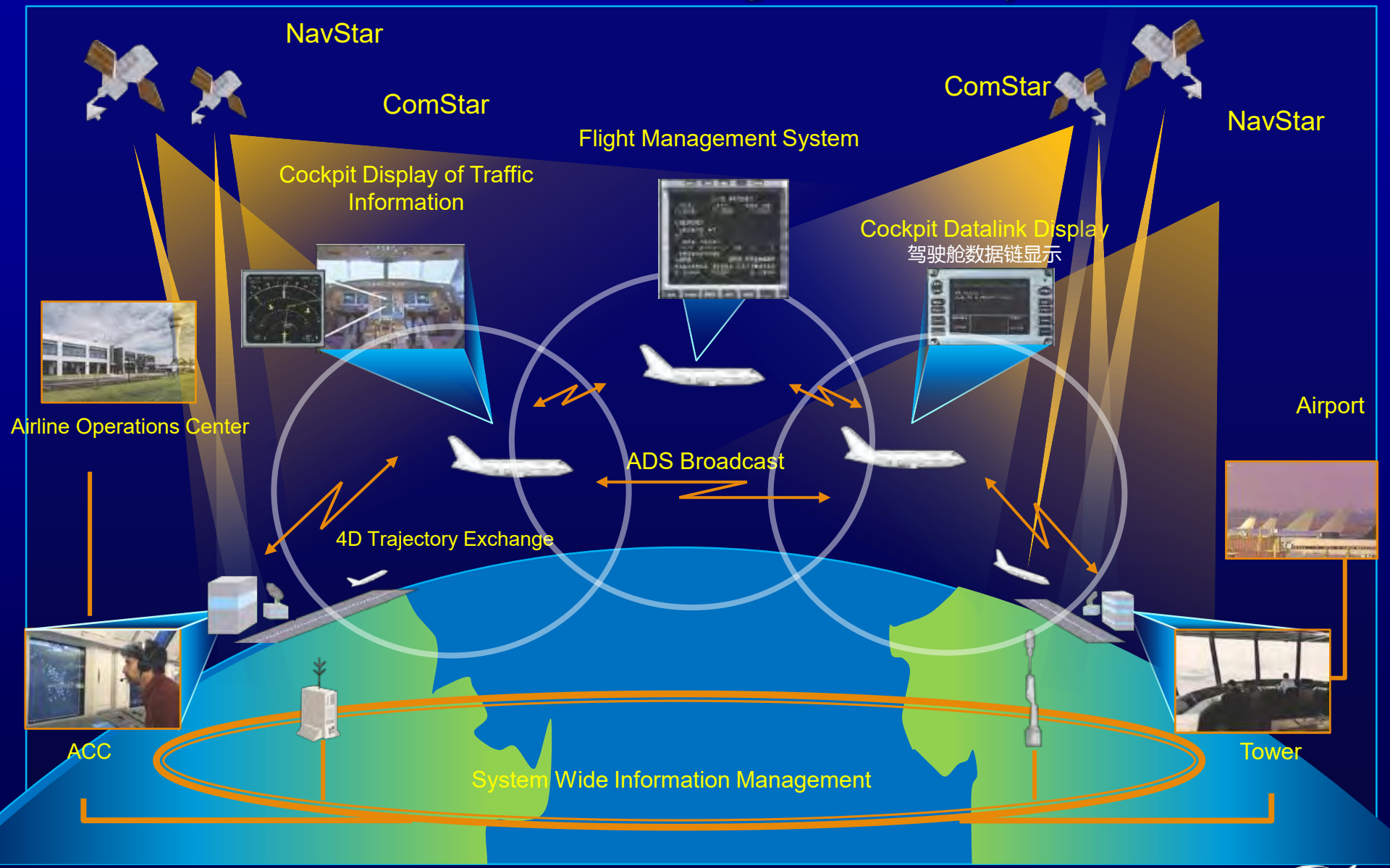
Utility System



System of Systems (SoS)



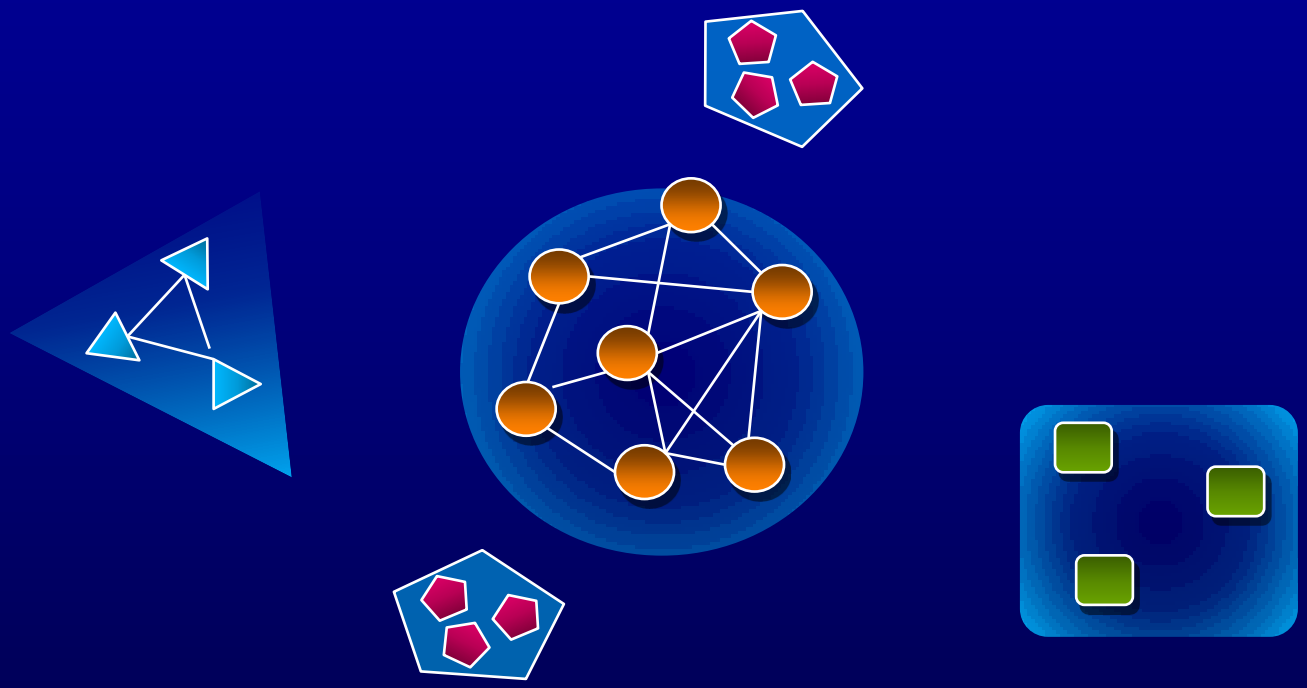
The New Air Traffic Management System



Cooperative Flight Data Processing System and traffic flow management

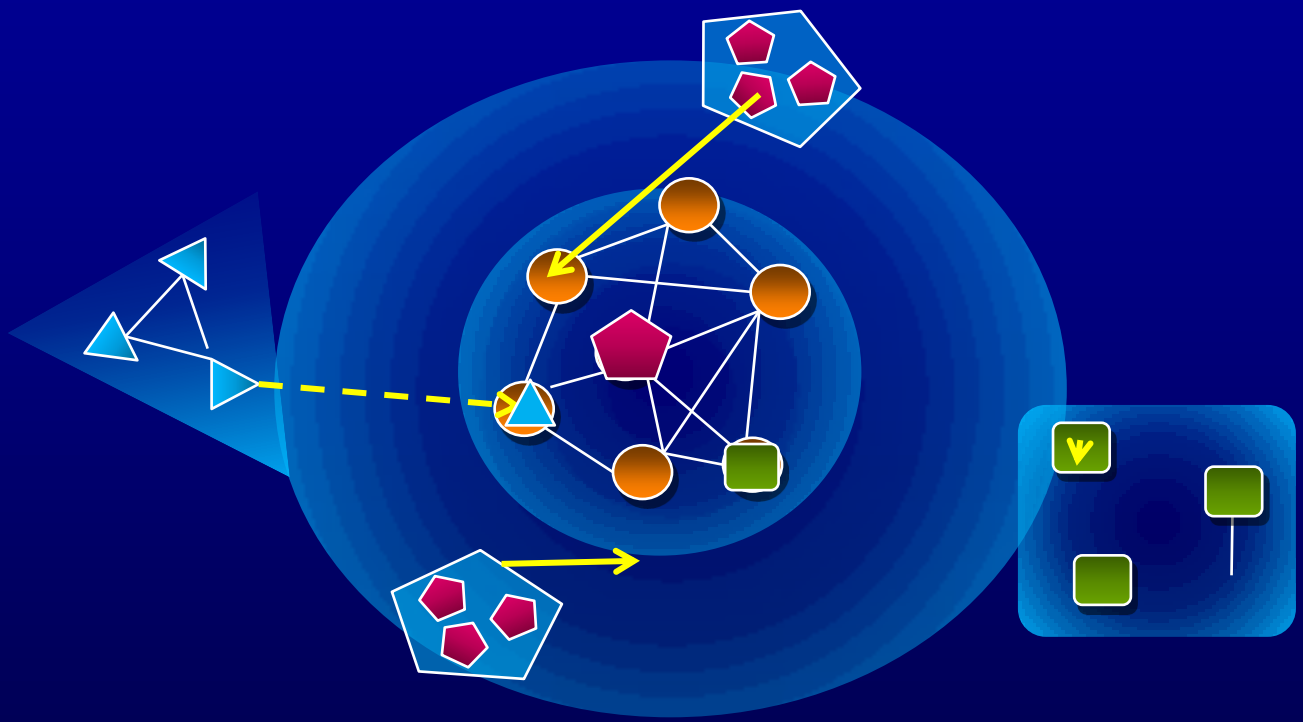
Evolution of Technology Complexity

___ Continuous Replacement and Enhancement



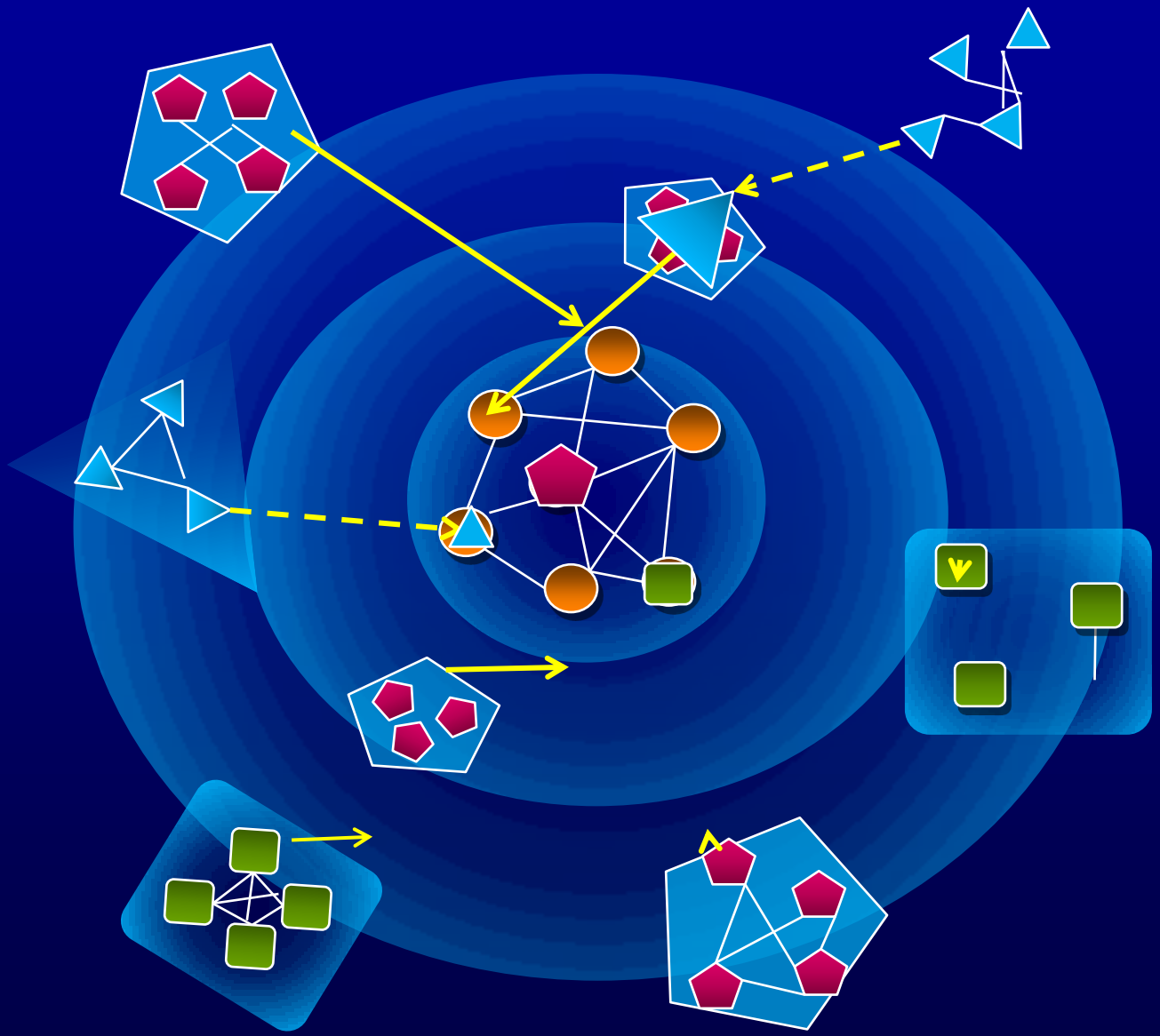
Evolution of Technology Complexity

___ Continuous Replacement and Enhancement



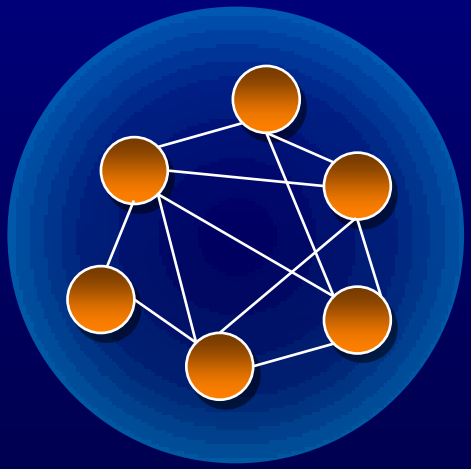
Evolution of Technology Complexity

___ Continuous Replacement and Enhancement



Evolution of the System Complexity

Mechanical System



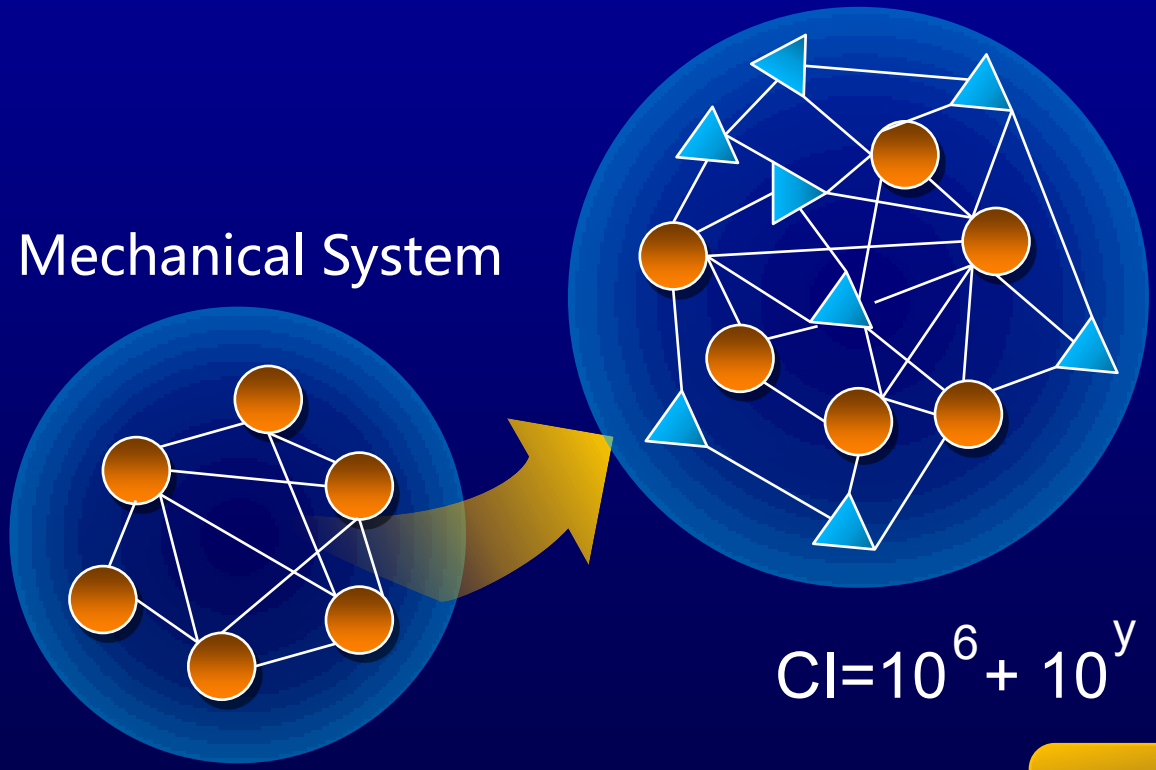
$CI = 10^{4} + 10^{x}$

$$CI = N_P + (N_{PIR} + N_{SER})$$

Evolution of the System Complexity

Mechatronic System

Mechanical System

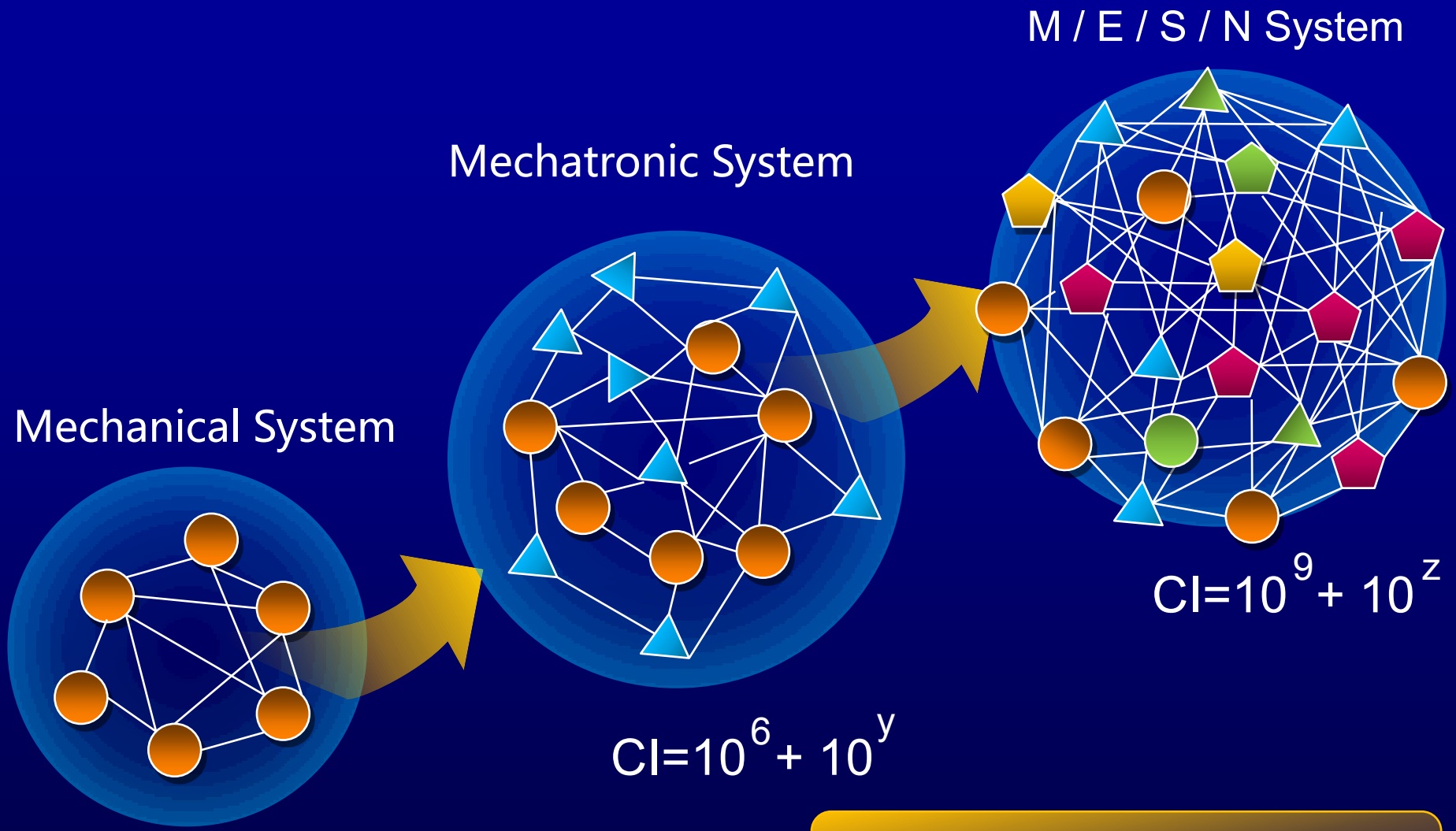


$CI=10^{6}+10^{y}$

$CI=10^{4}+10^{x}$

$$CI = N_P + (N_{PIR} + N_{SER})$$

Evolution of the System Complexity



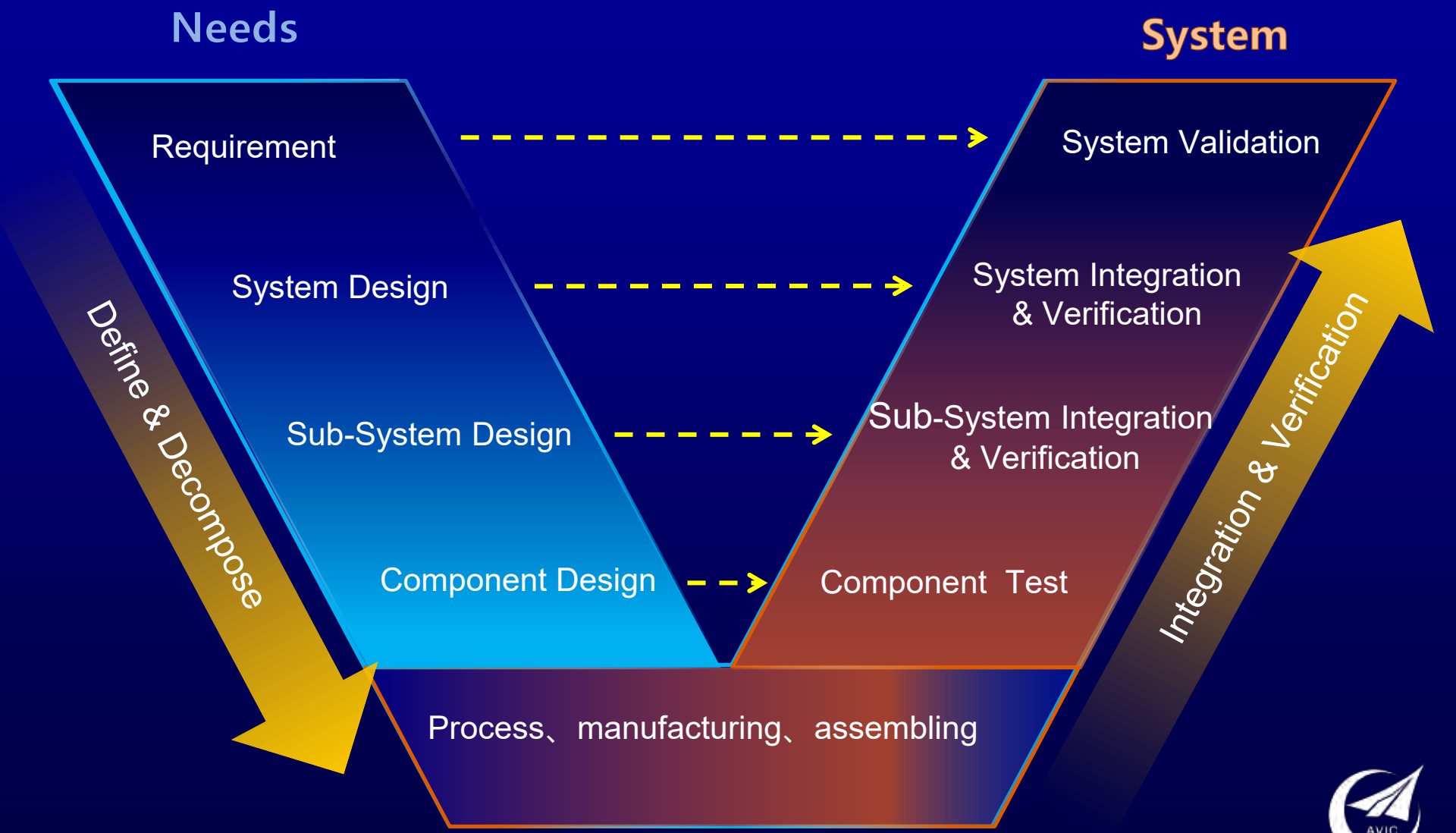
$$CI = N_P + (N_{PIR} + N_{SER})$$

Evolution of Aeronautic Complex System

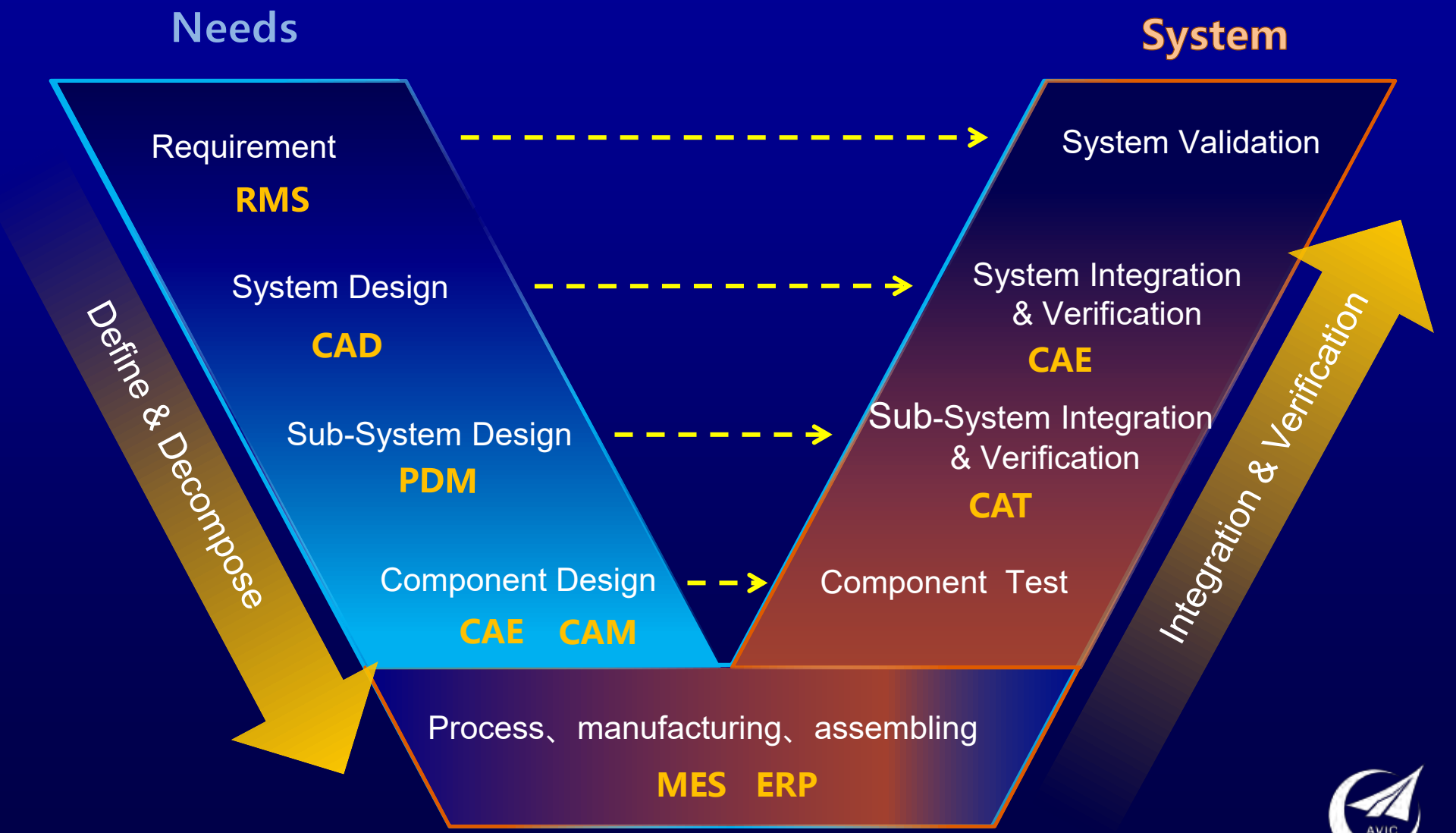
Challenge to Traditional System Engineering

Evolution of Complex System Engineering

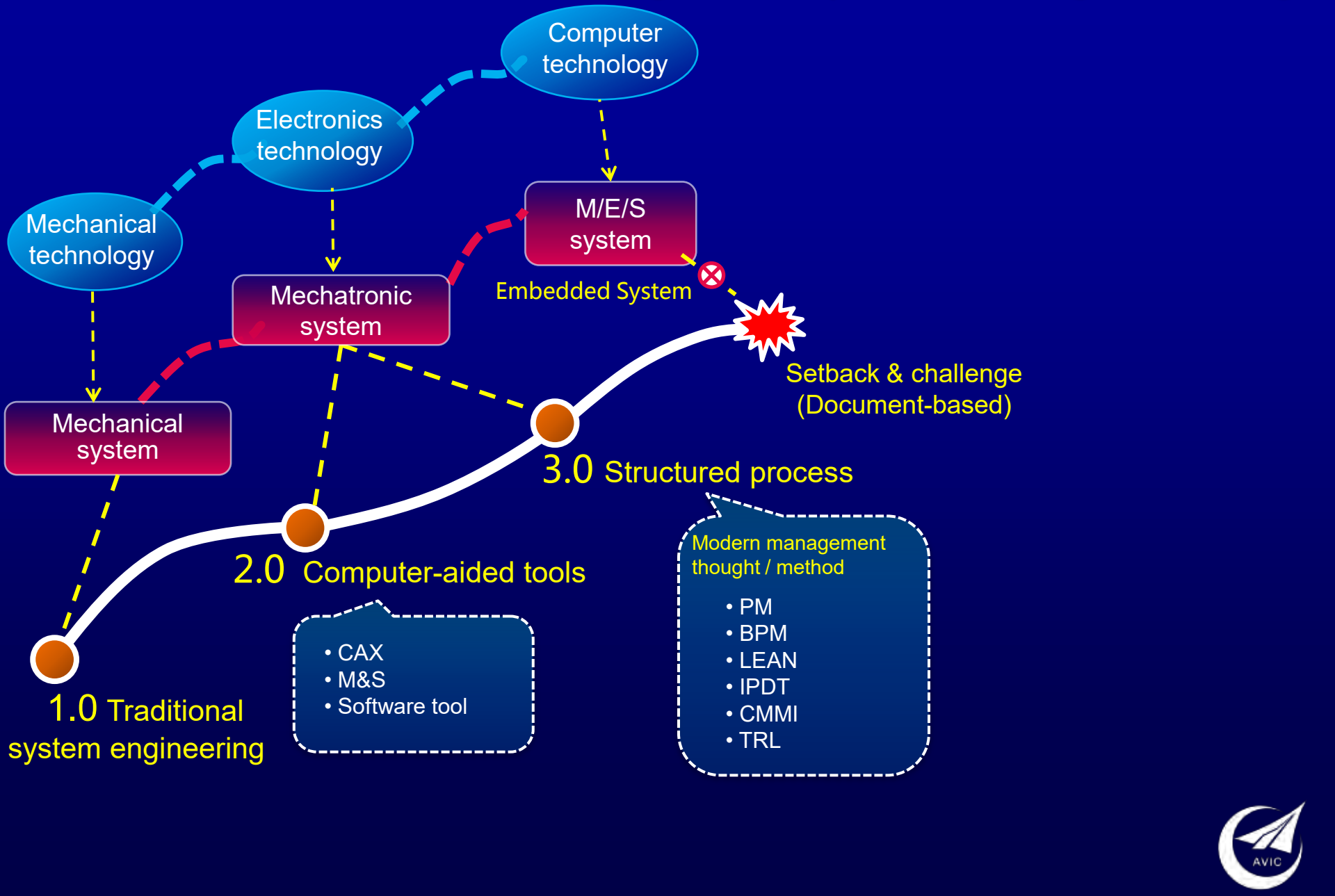
"V" System Engineering Process



Computer Aided Tools to Support SE Activities



Evolution of and Challenge to System Engineering

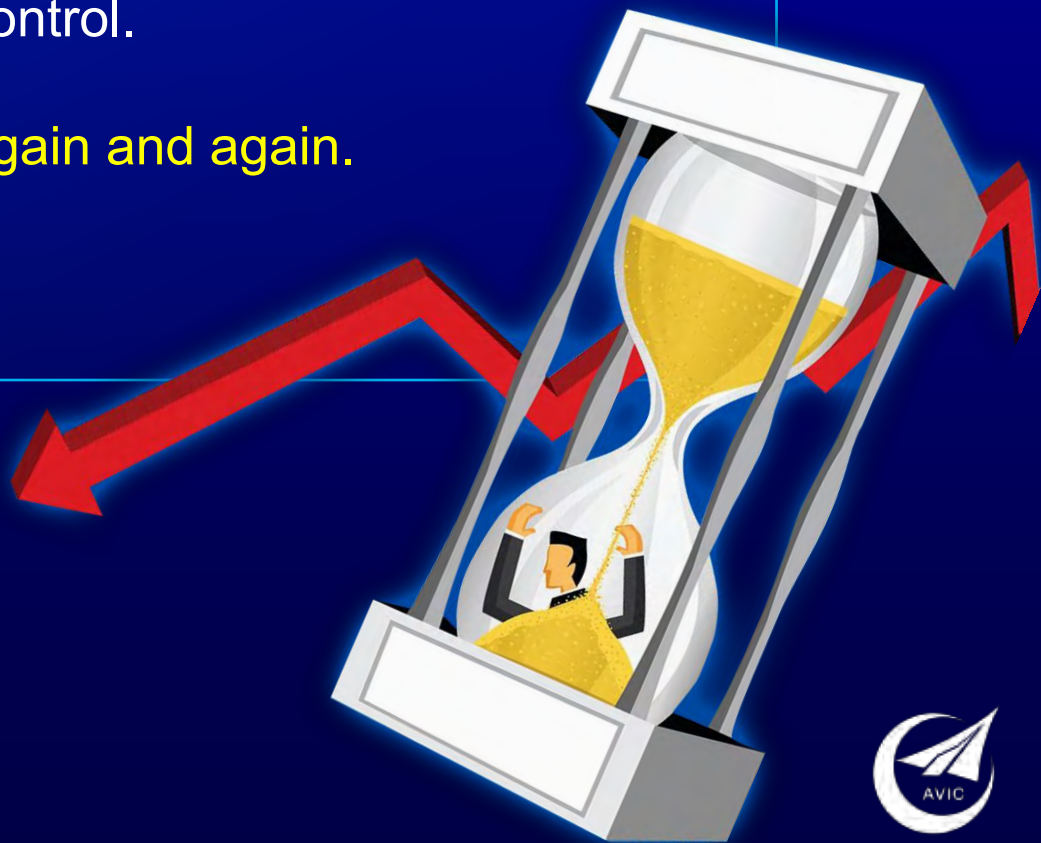


Why did Programs Fail Repeatedly?

Many high-end product programs in the aerospace industry have been postponed, cost overrun.

These highly complex programs with high-risk are system engineering with strict process control.

So, why these programs failed again and again.



What is the Root Cause?

The main reason for program failure lies in the interaction of the unexpectedness and unpredictability among elements in the system.

Problems will not be exposed until the system integration or test stage, sometimes even worse, until the delivery to users.

More interactions of systems, more serious the problem is.

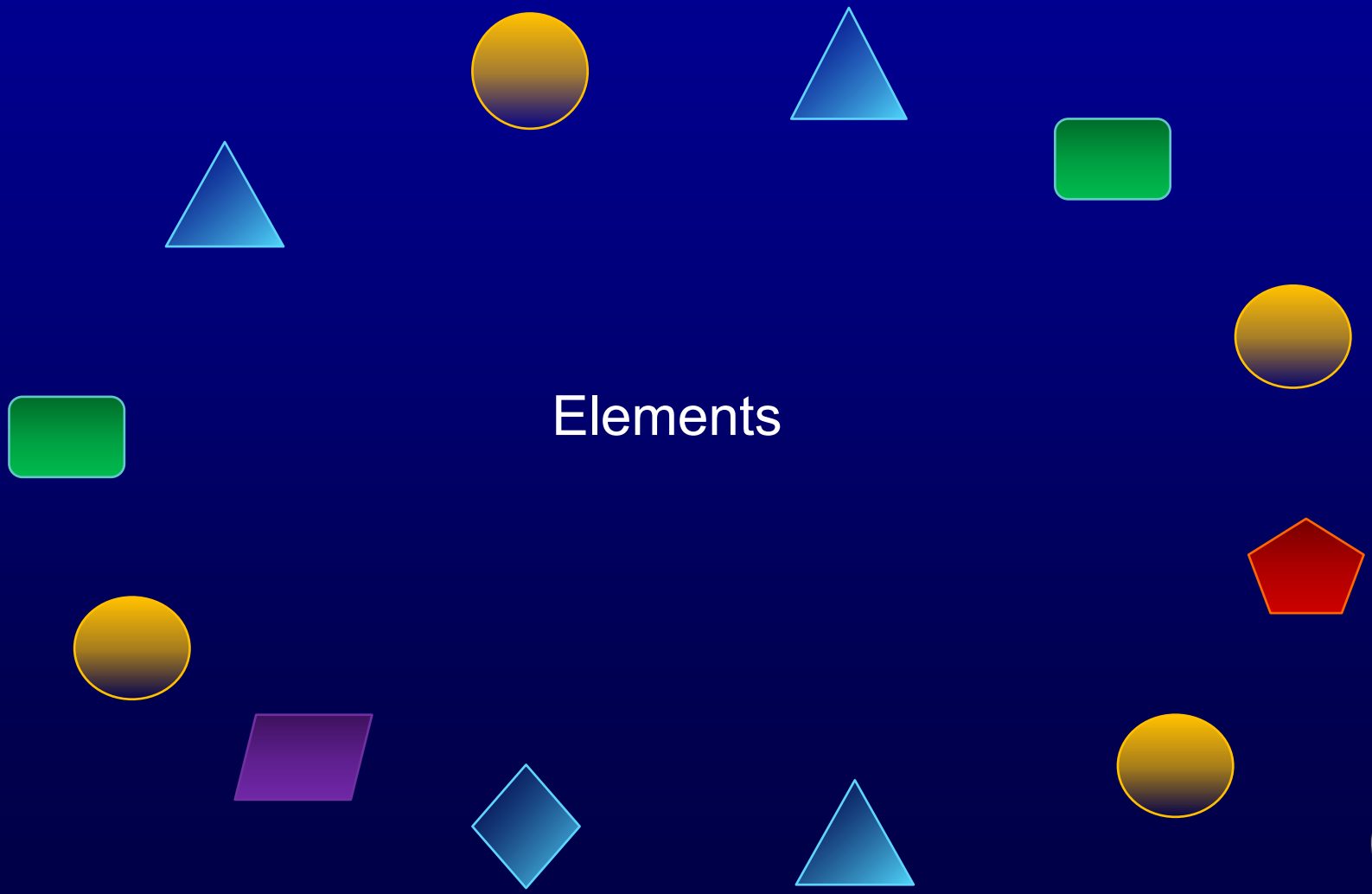
Problems and Consensus

Problems , often occurs at the interface of elements in the system, most of which are among independent elements that are considered to have been decomposed, esp. in software and communication.

Consensus , apparently current system engineering can not meet the requirements of the growing, large, comprehensive and complex industrial system.

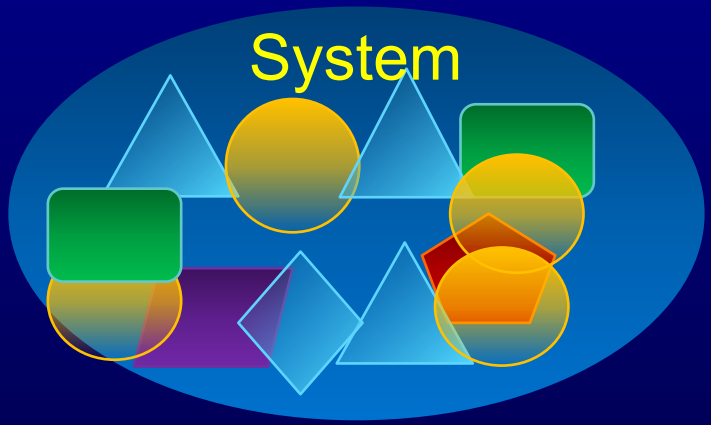
Emergence - The Key Characteristic of Complex Systems

To understand and estimate the interaction among functions.
‘Emergence' is a challenge and nightmare
Murphy’s law always works.



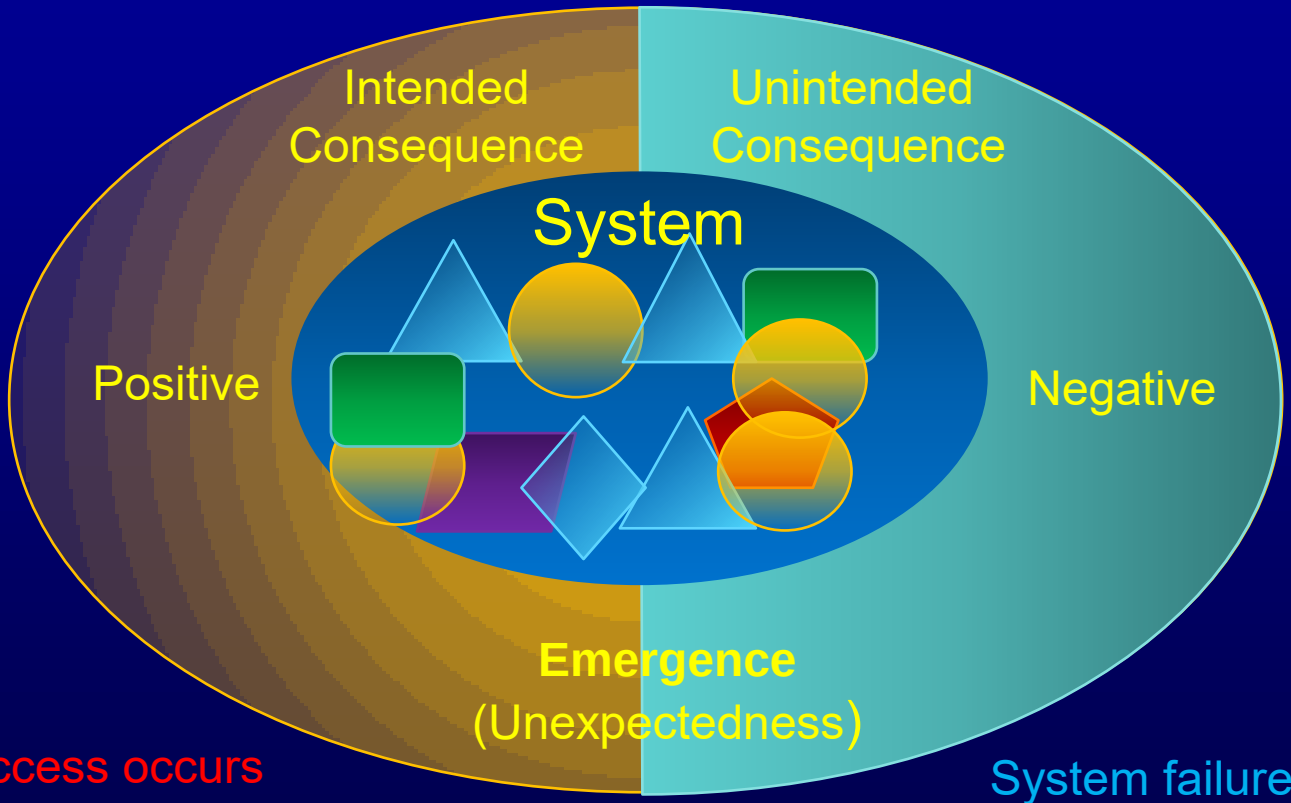
Emergence - The Key Characteristic of Complex Systems

To understand and estimate the interaction among functions.
‘Emergence' is a challenge and nightmare
Murphy’s law always works.



Emergence - The Key Characteristic of Complex Systems

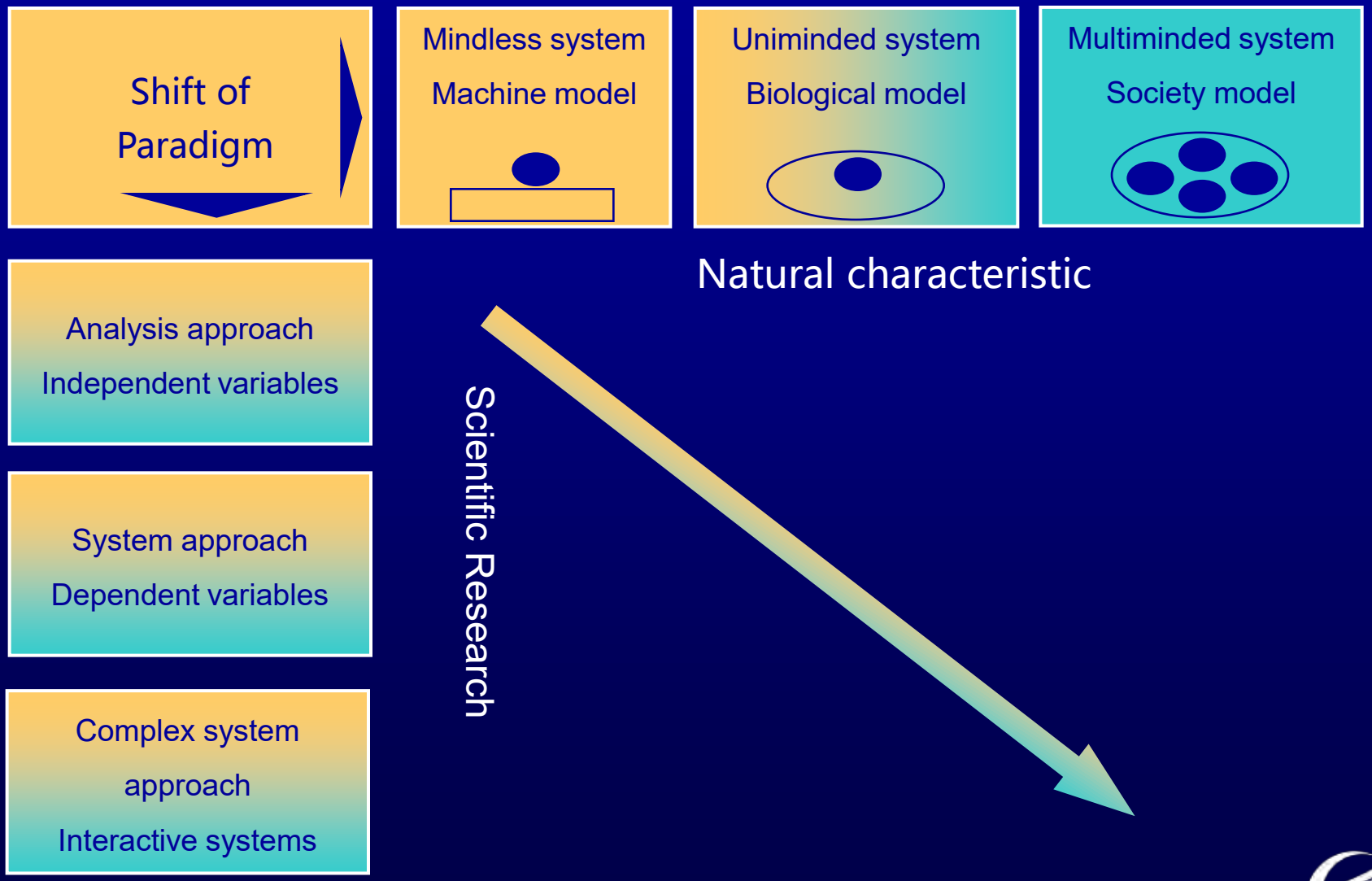
To understand and estimate the interaction among functions（not Physical）.
‘Emergence' is a challenge and nightmare
Murphy’s law always works.



System success occurs
When the anticipated
Properties emerge

System failure occurs
When the unanticipated
Emerge properties appear

Dual Paradigm Shift

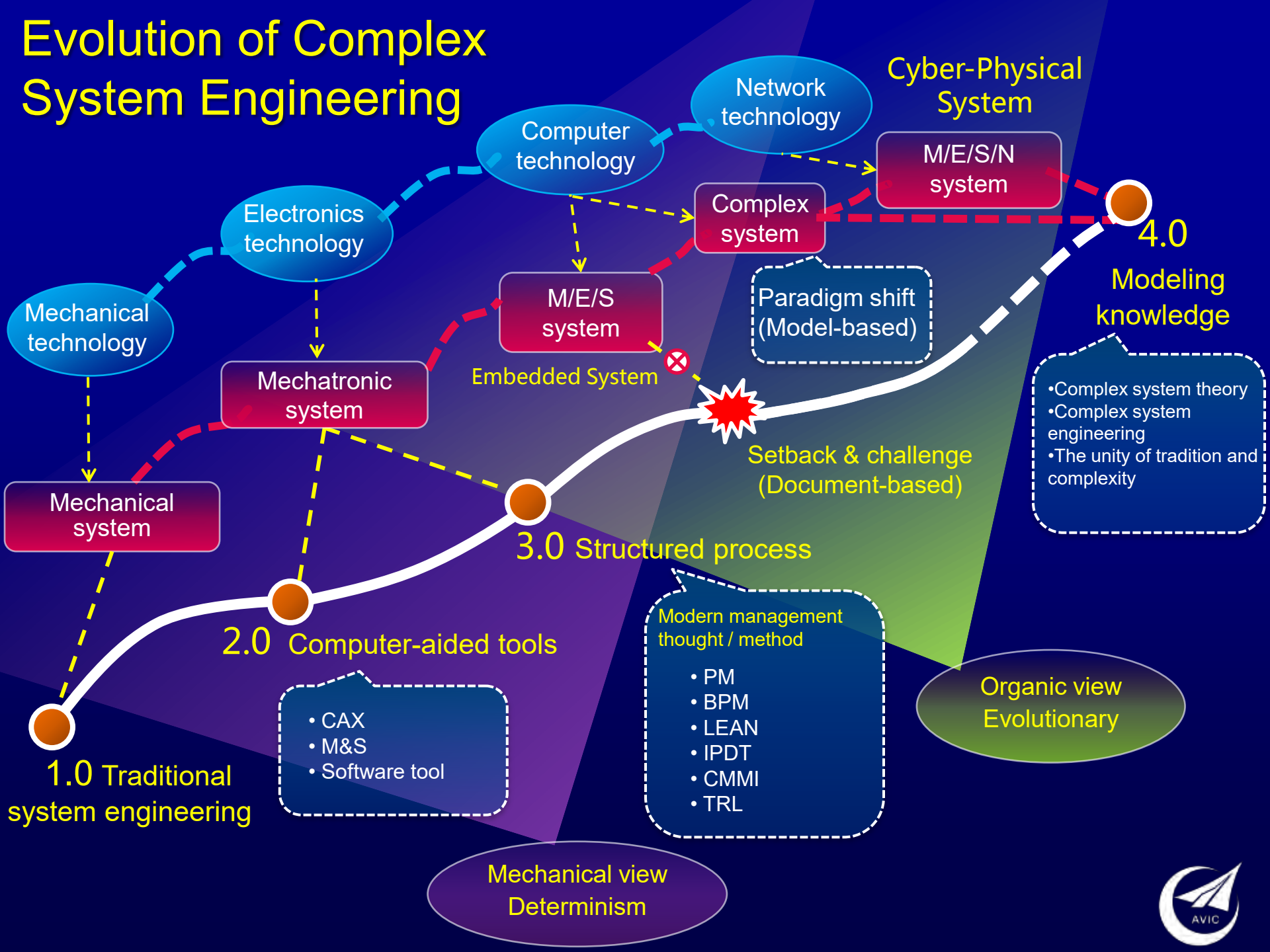


Evolution of Aeronautic Complex System

Challenge to Traditional System Engineering

Evolution of Complex System Engineering

Evolution of Complex System Engineering



Transform to MBSE

Past

SE Artifact
- Specification
- Interface requirement
- System design
- Analysis & Trade-off
- Plan test

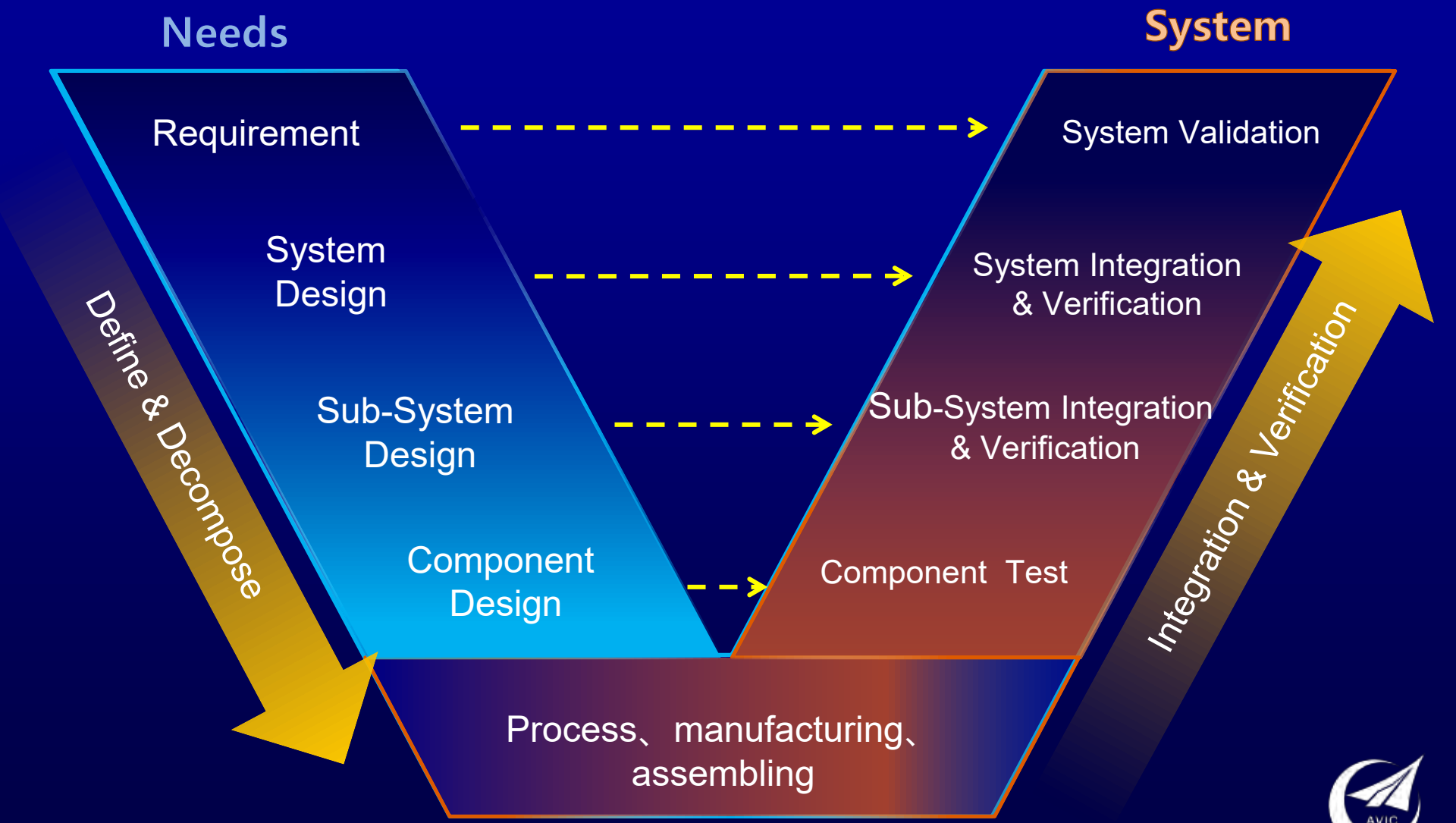
Future

Document - centered

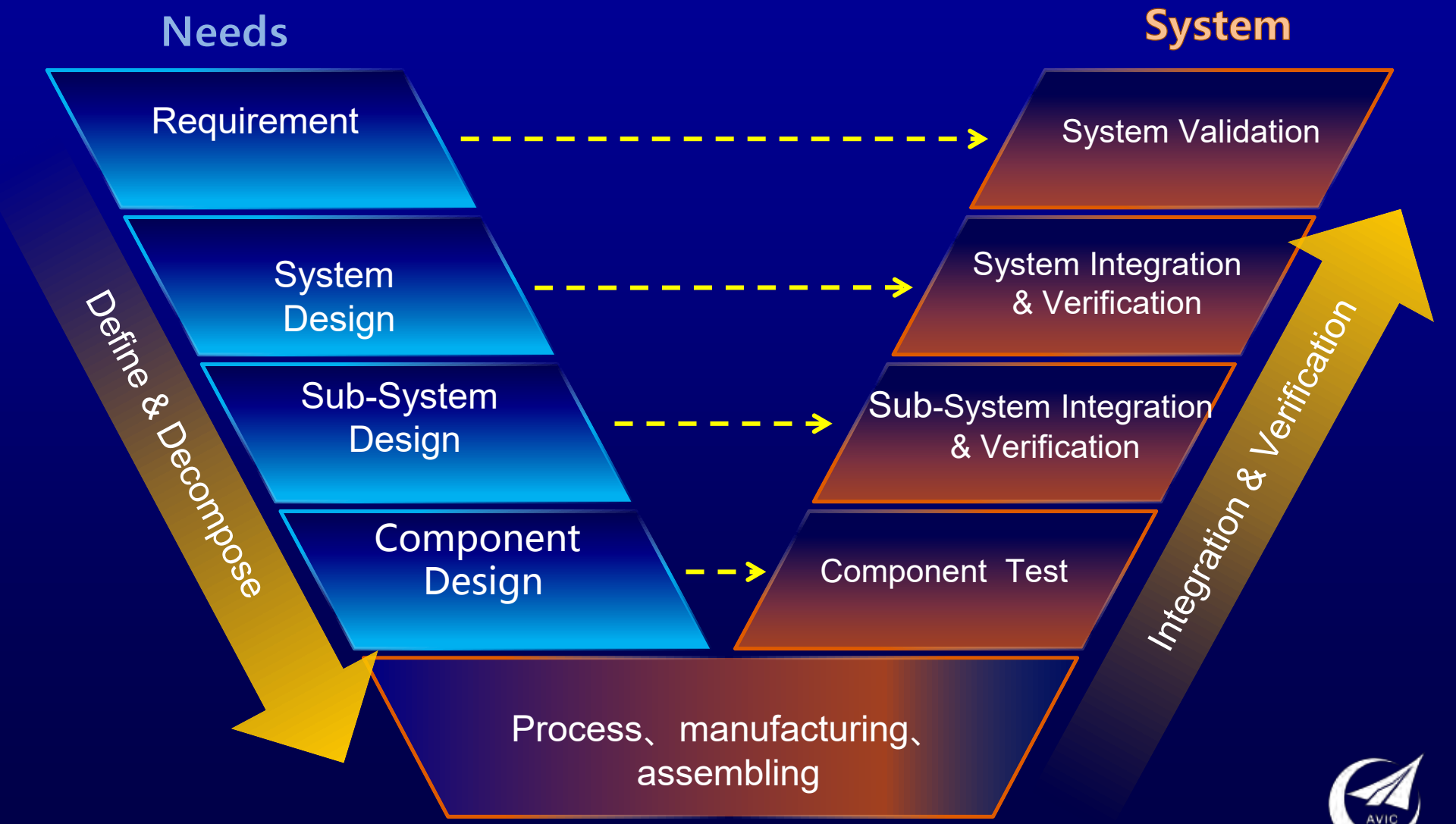
Model - centered

Model, not document, is the main output or artifact of System Engineering .

"V" Process of System Engineering

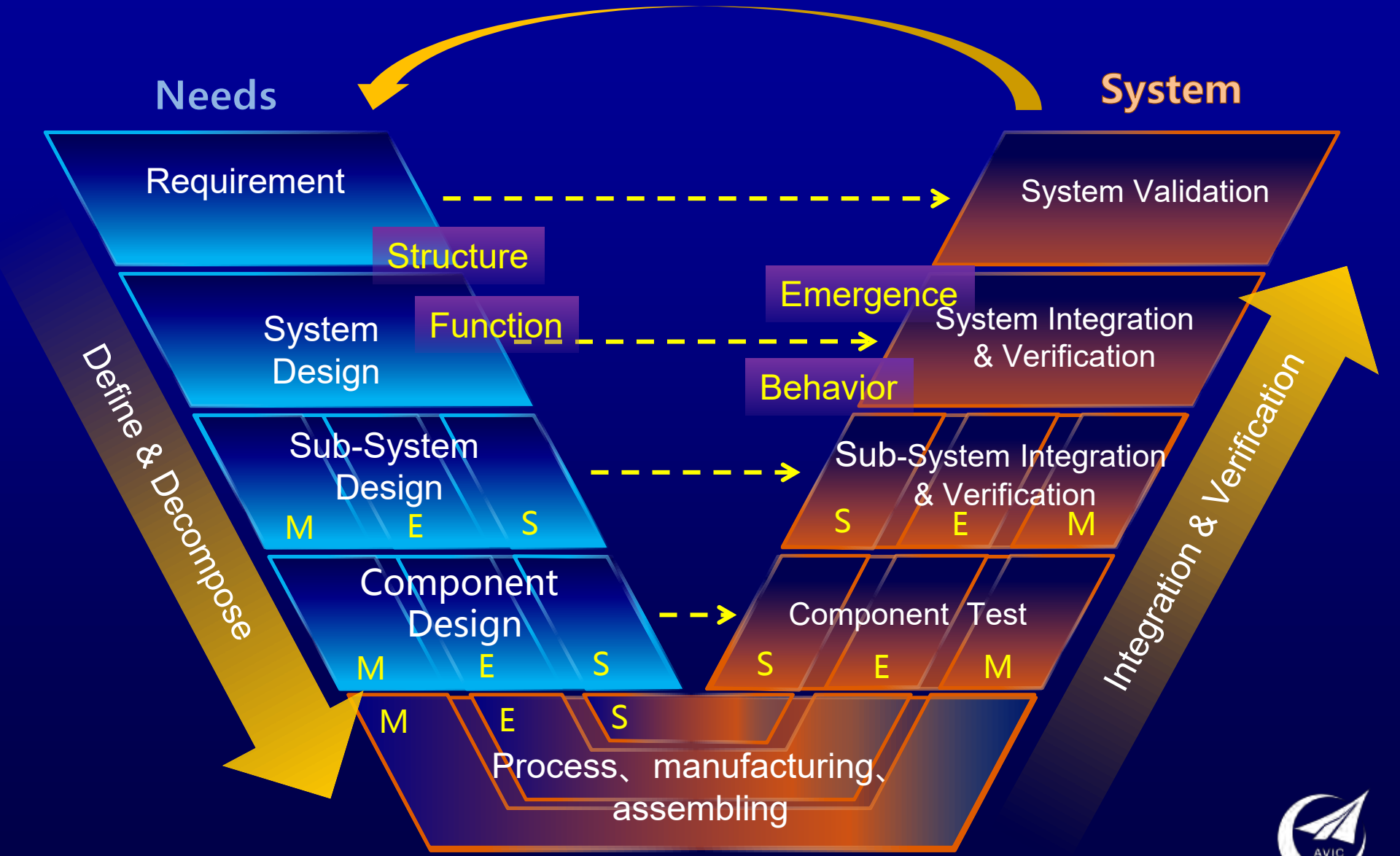


"V" Process in Reality



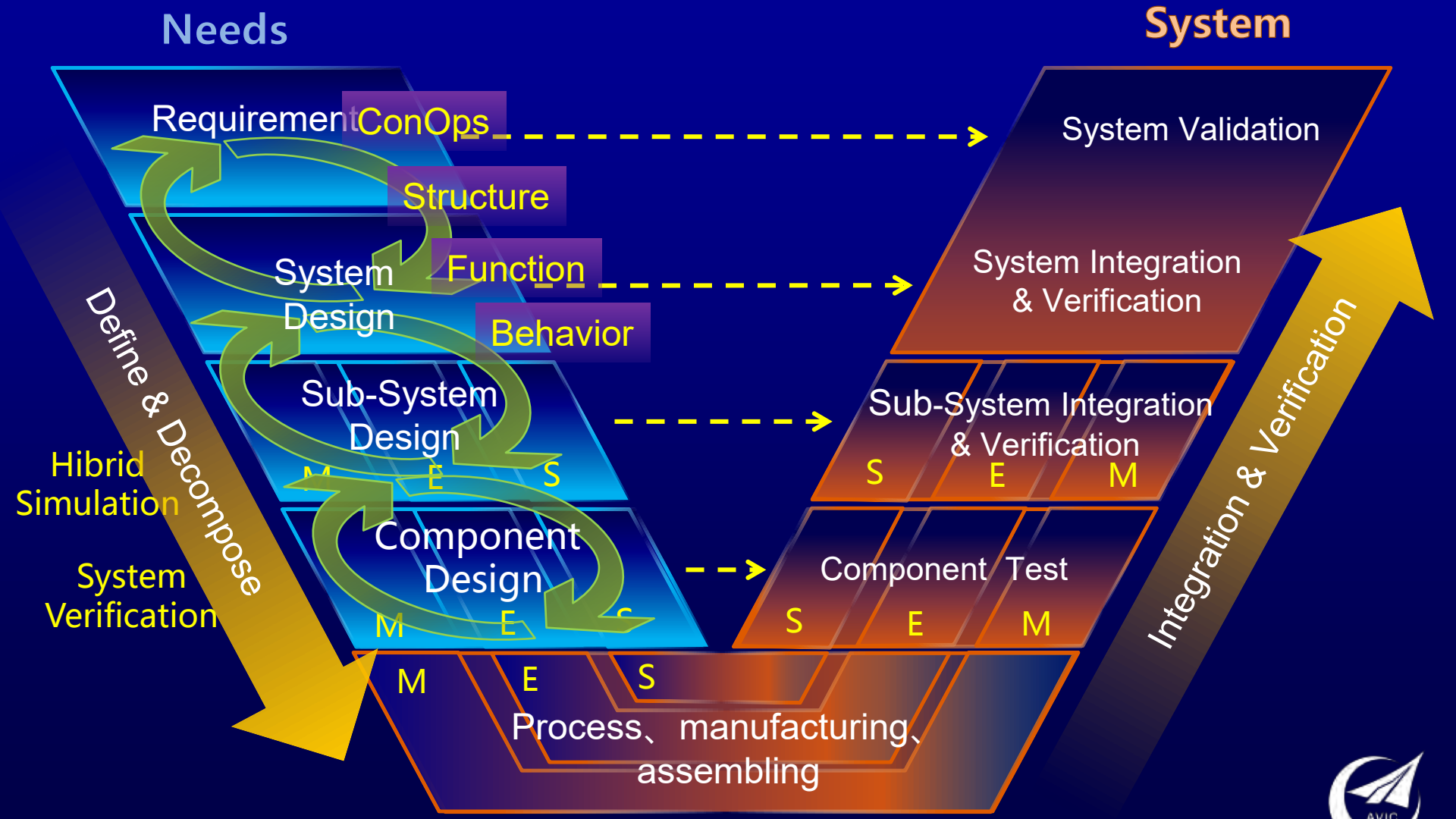
"V" Process in Reality

System Behavior appears after Integration

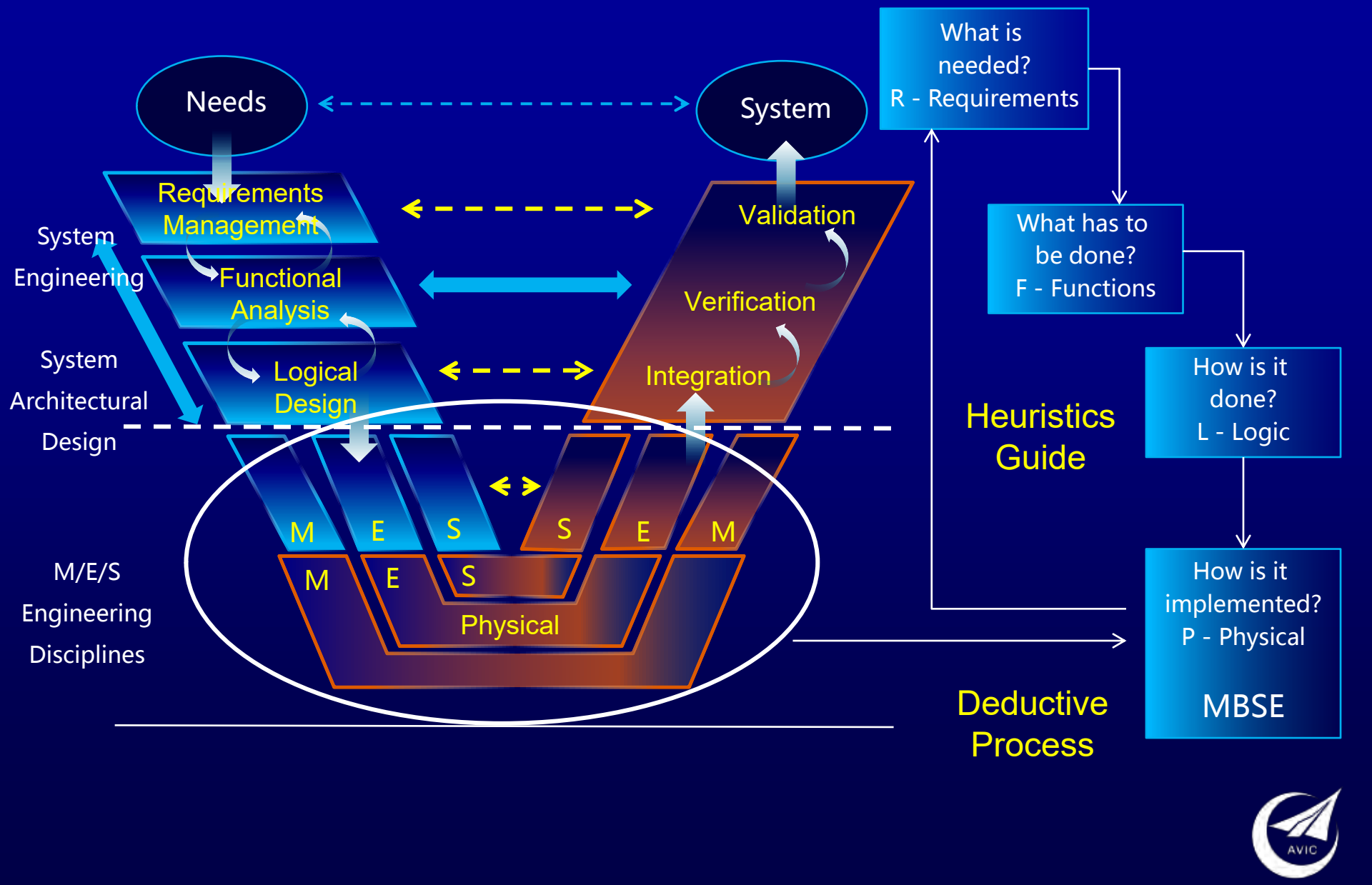


Continuous Verification of MBSE

System Behavior appear Before Decomposition

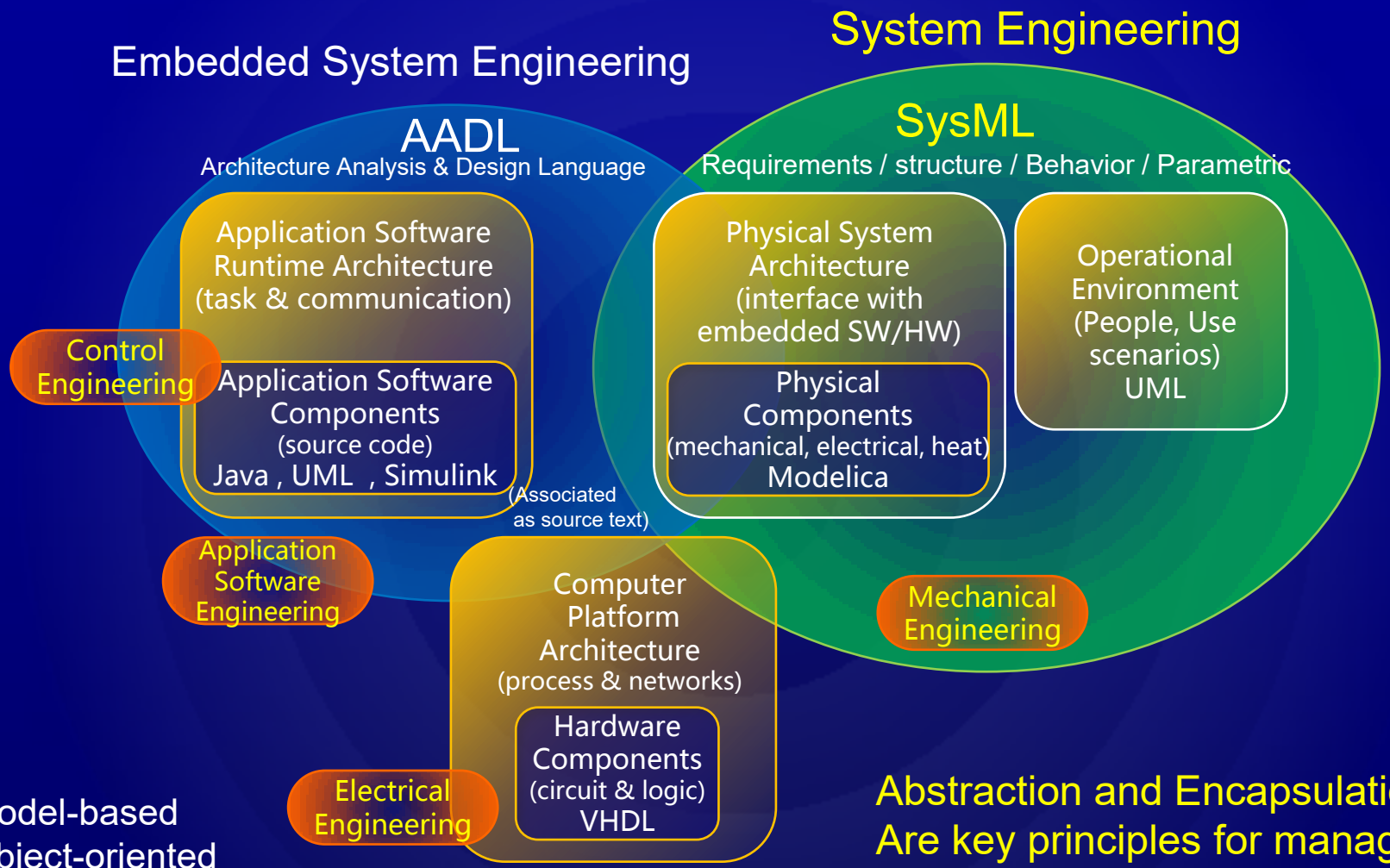


MBSE —— Support Complex System Design



Model_Based Engineering

Combination of Modeling Capabilities

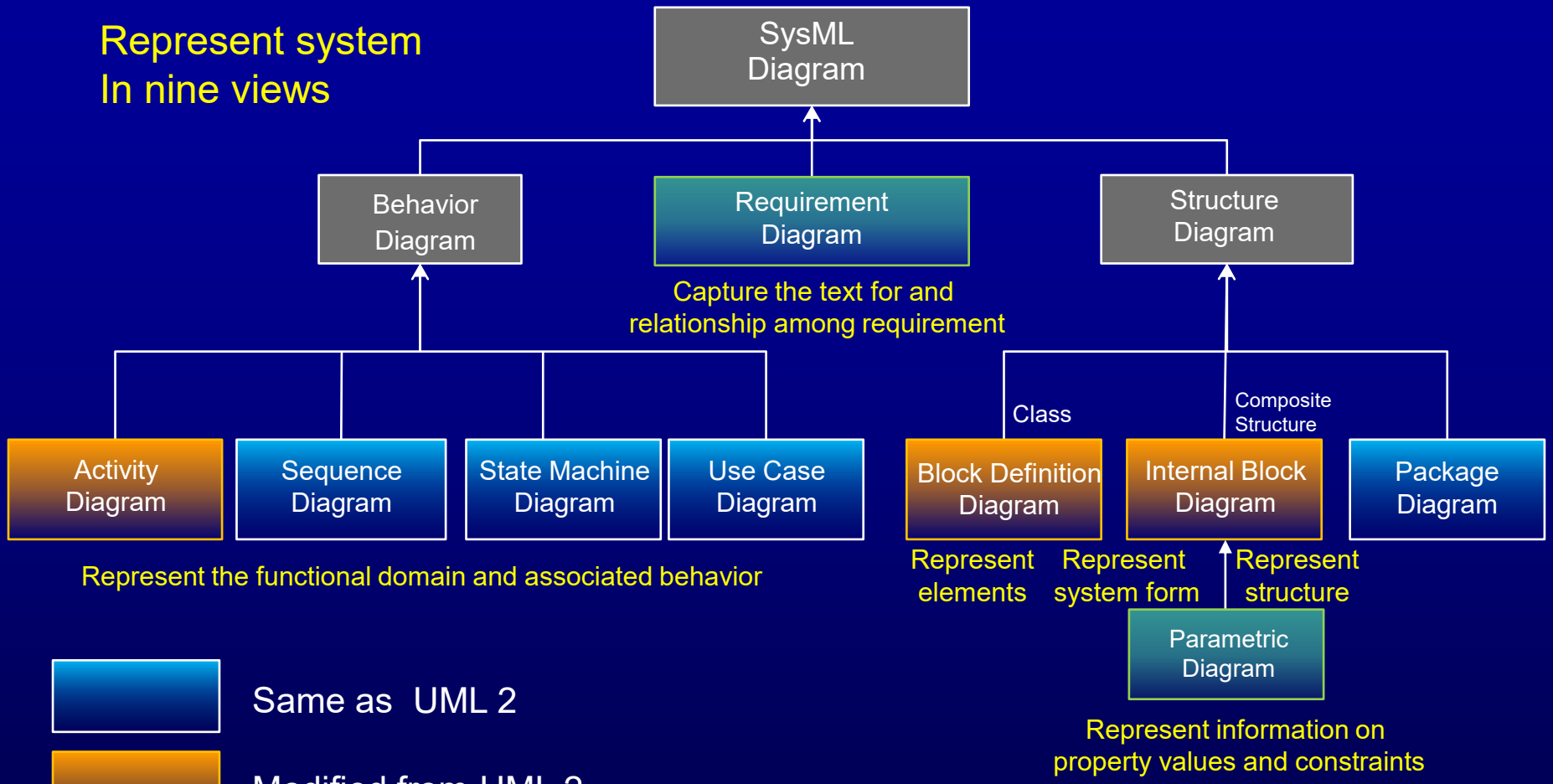


Model-based
Object-oriented
Model-Driven Architecture
Model-Based Engineering

Abstraction and Encapsulation
Are key principles for managing
The complexity of systems

SysML / UML

Represent system
In nine views

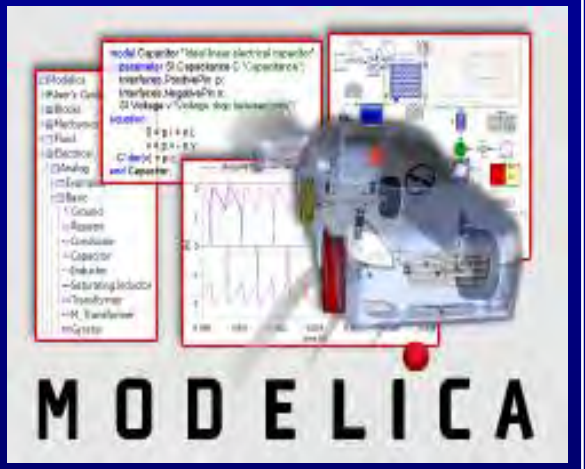


Two new Open Standards

Modelica and the Modelica Association

Modelica is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents.

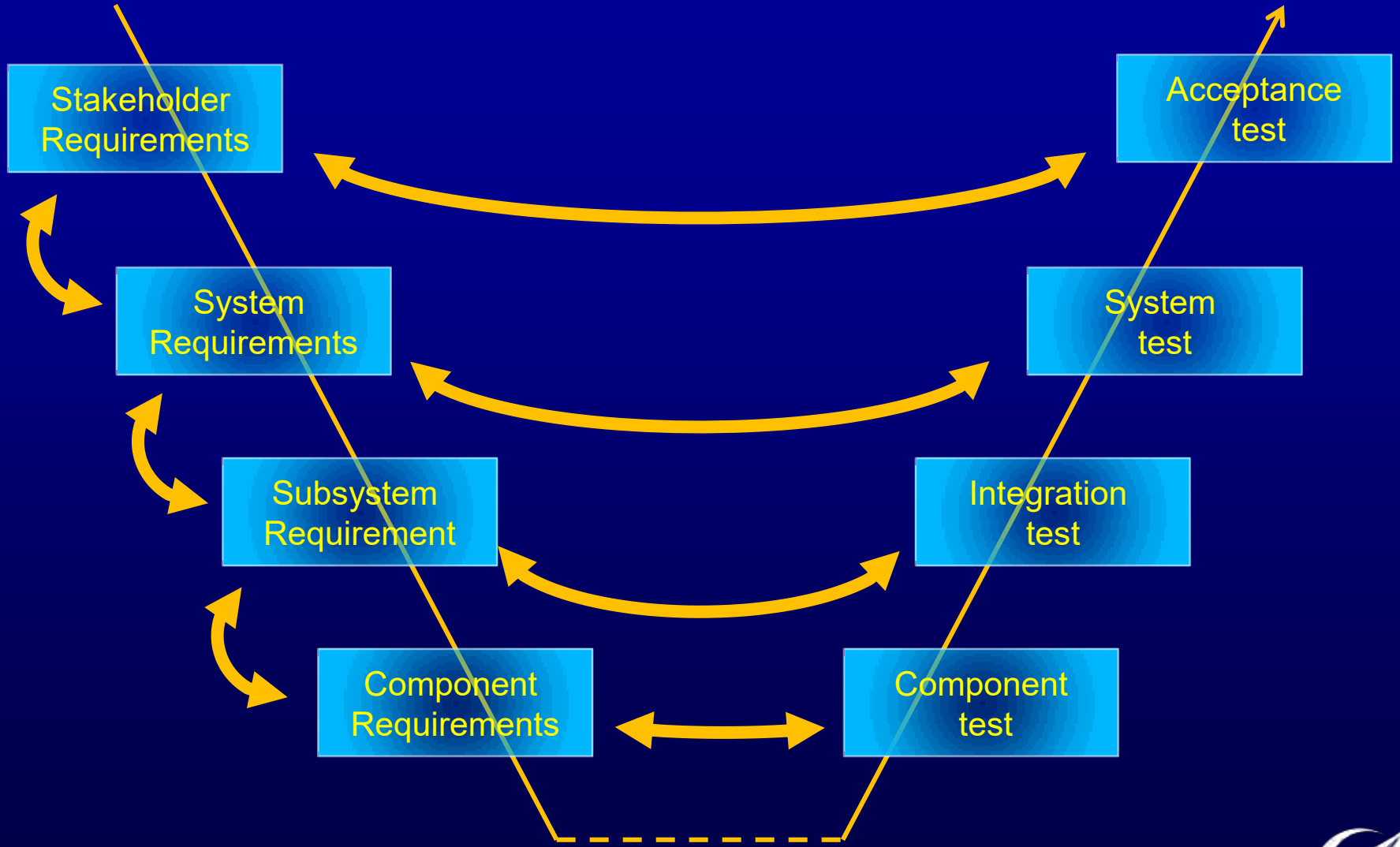
Modelica Libraries with a large set of models are available(overview, details and download). Especially, the open source Modelica Standard Library contains about 1280 model components and 910 functions from many domains.



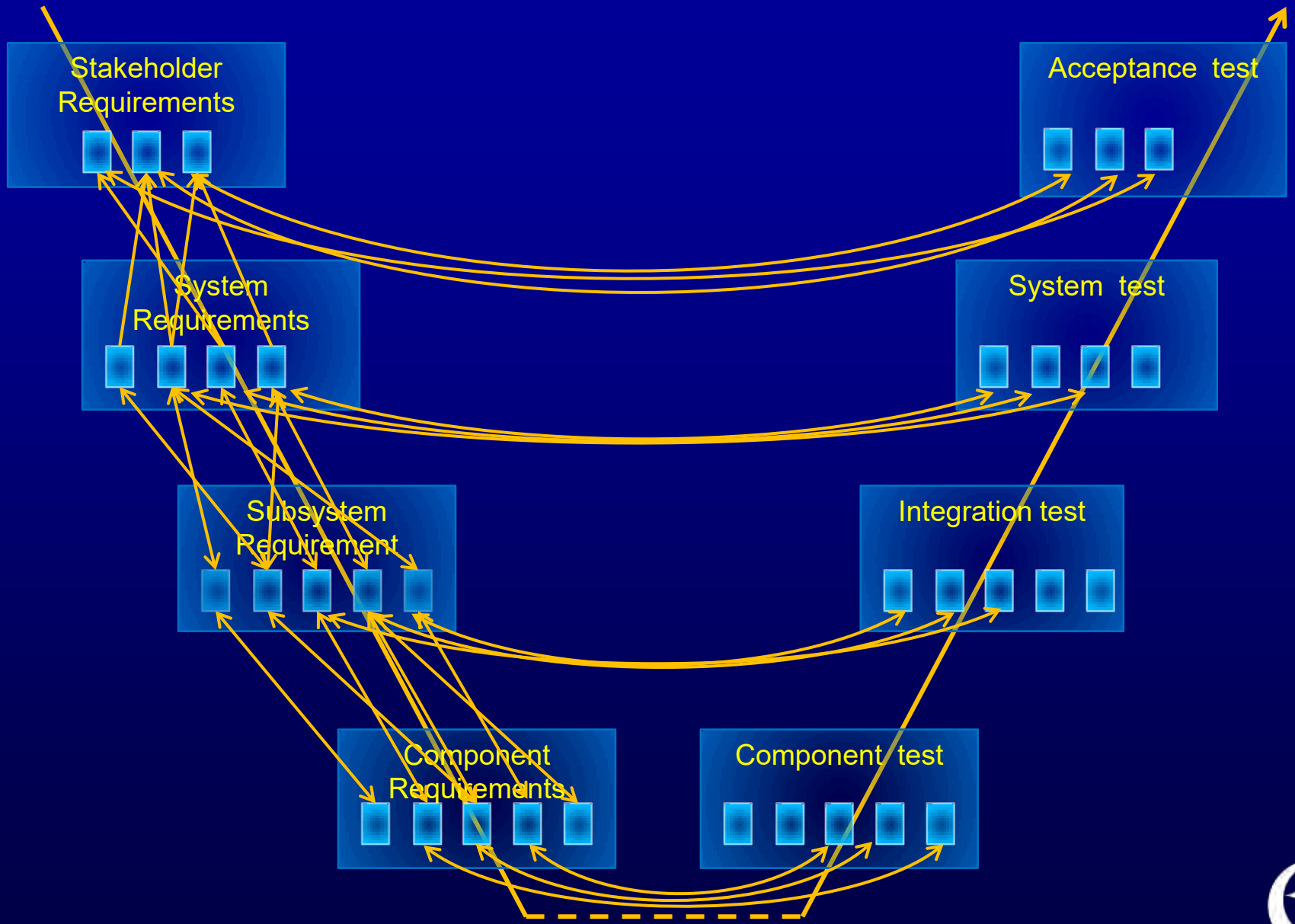
Functional Mock-up Interface

Functional Mock-up Interface(FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models

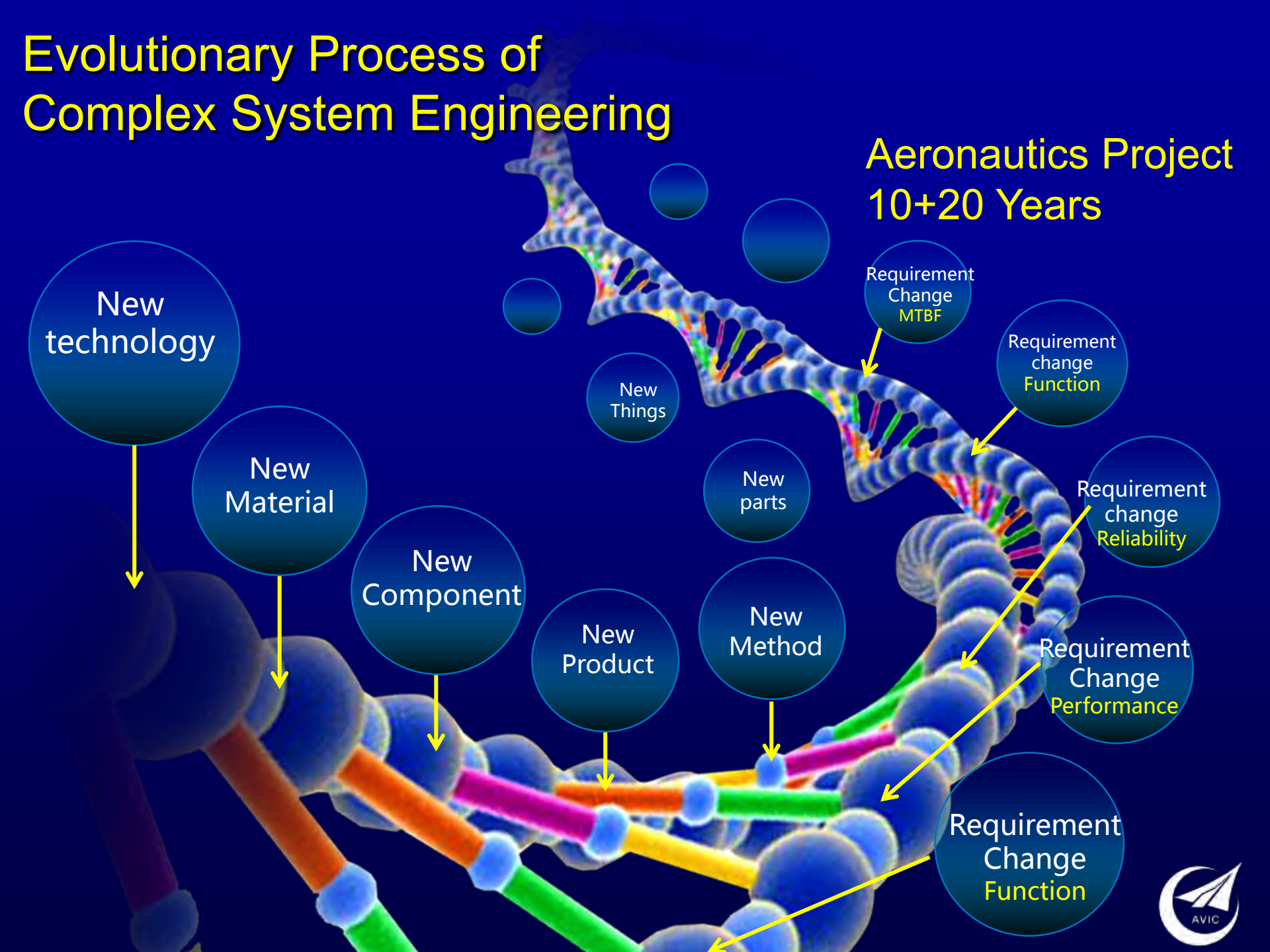
Continuous Verifiability of Requirement



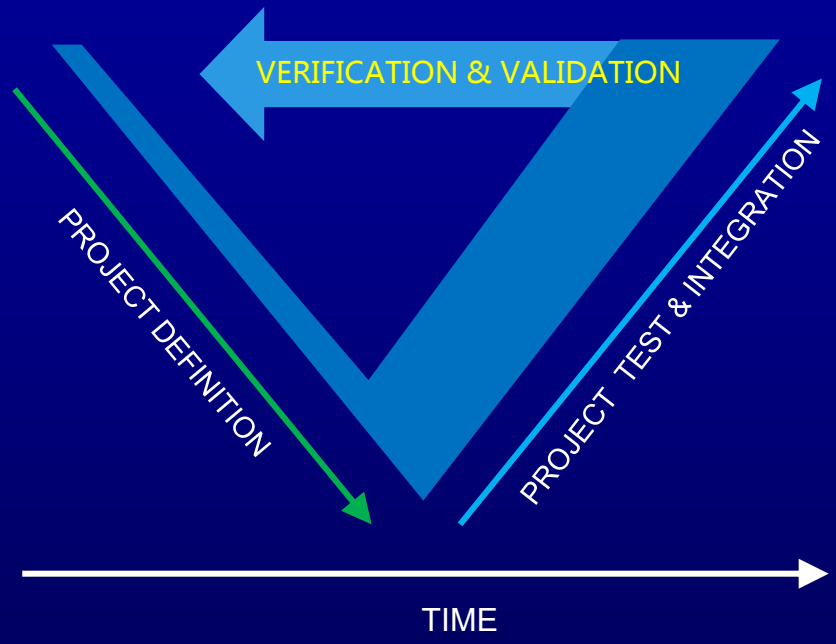
Traceability of Requirements

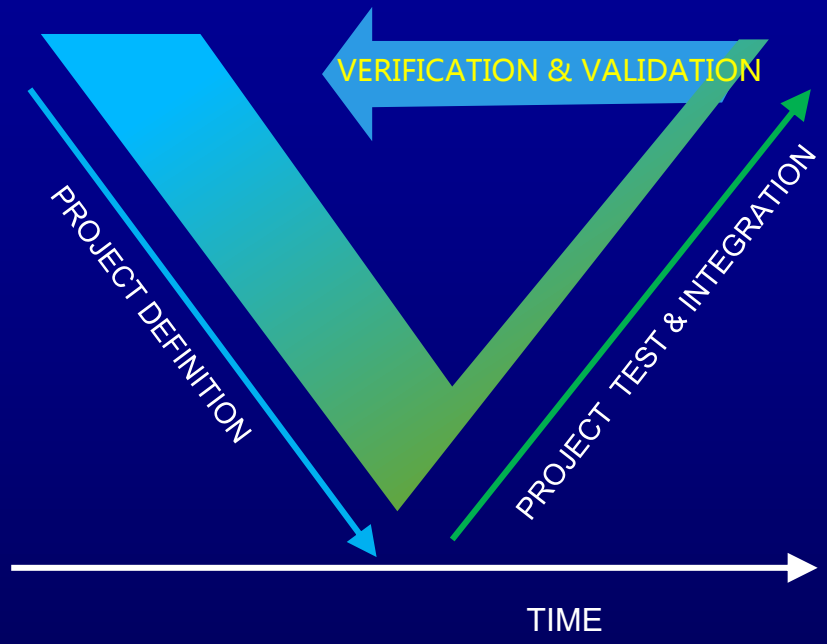


Evolutionary Process of Complex System Engineering



Change to MBSE





Now

Future

Strategy to Deal with Complexity

Disciplined Approach ——
Structured Process Knowledge

Deal with Uncertainty of Complex System

Systemized Model ——
Structured Element Relations

Cope with Evolution of Complex System

Thank you