The Challenge of Risk Assessment In Complex Information Systems

Dr. Robert Bonneau

Need for Complex Systems Risk Analysis

Problem: There is no routine way to assess risk for information services today

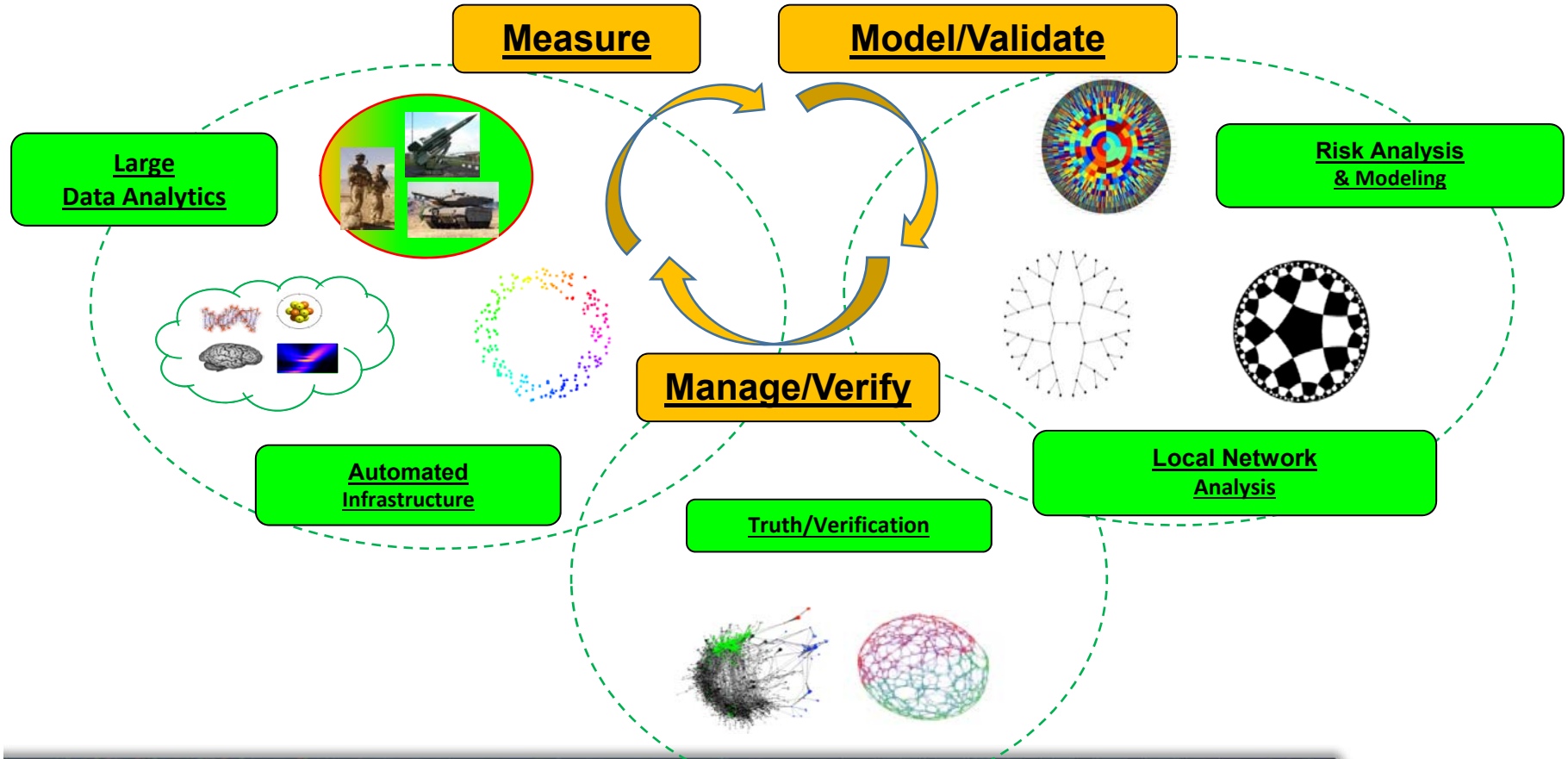
- **AI risk analysis is in its infancy**
- **Cyber risk is treated as isolated activity**
- **Infrastructure risk is deferred to commercial providers**
- **Many humans are in system with varying degrees of performance risk**

Solution: Create system models and introduce into model driven software engineering framework as a part of a larger process

- **Include AI algorithm labelling and training as part of automated model driven software architecture**
- **Modelling of cloud and embedded data driven processes**
- **Automate refactoring of information services**

Measure/Model/Manage

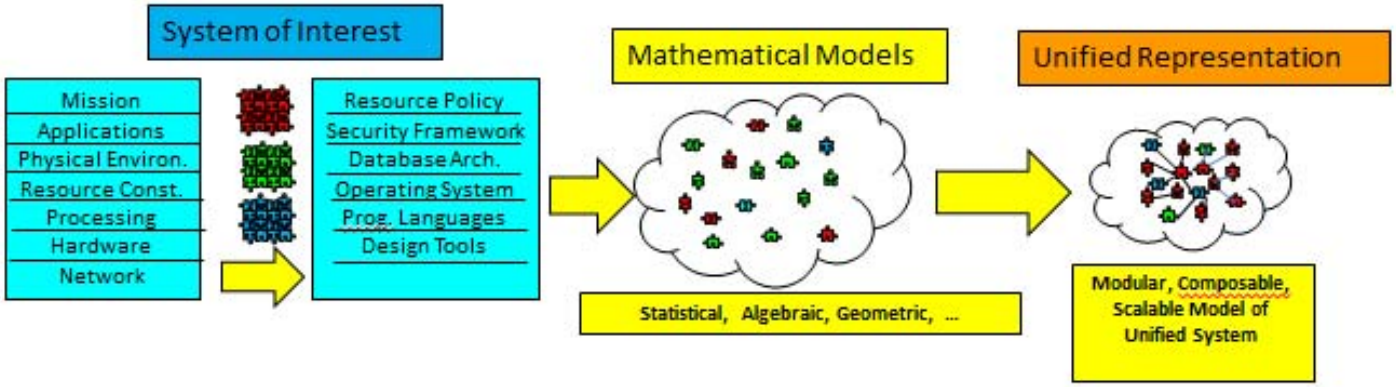
Integrated modeling, validation, verification, and management can characterize mission performance with advanced data models



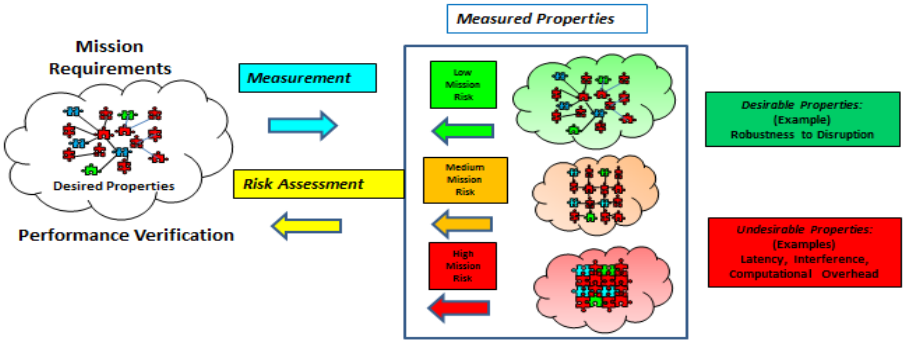
Measurement

We wish to understand how to measure the state of a mission on an infrastructure

What to measure?



How to measure?



Measurement

- We have some signal environment S and interference environment E and we wish to measure these with some measurement vector Φ_Λ

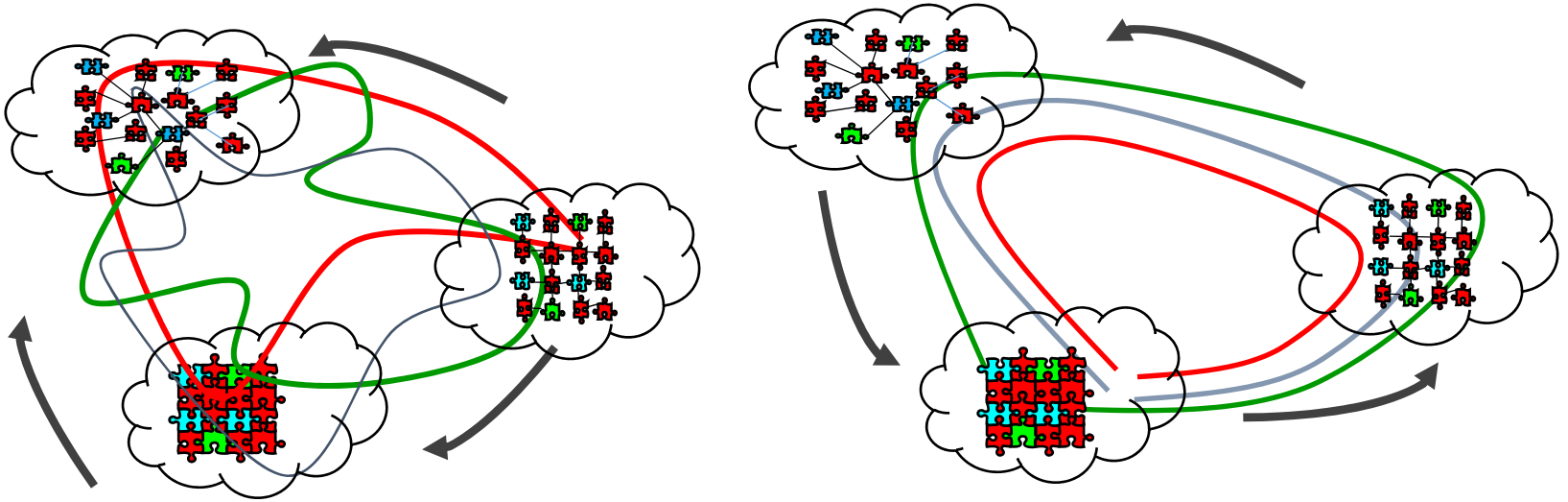
$$X = \Phi_\Lambda \otimes (\mathrm{S} + \mathrm{E})$$

The problem in most measurement scenarios is getting the right Φ_Λ to separate S from E. We define our measurement process such that Φ_Λ consists of the individual projections on the environment defined by $\phi_{i,i\in 1,\ldots,N}$.

Modeling

We must have validated models of mission performance which can come from known models or empirical data

Mission Operation Trade-space



Un-validated Modalities (high mission risk)

Validated Modalities (low mission risk)

- In a distributional context we will illustrate such a concept in terms of the mean vector, m and Covariance matrix of a normally distributed model, C_{xx}. Thus our mean and covariance matrices for our measured data from section II are indicated:.

$$m = E(X) \ , \ C_{XX} = E((X - \mathrm{m})(X - \mathrm{m}))$$

- Our approximated or un-validated covariance is indicated:

$$\tilde{C}_{XX} = E\left\{\begin{bmatrix} \tilde{c}_{11} & \tilde{c}_{12} & \cdots & \tilde{c}_{1N} \\ \tilde{c}_{21} & \tilde{c}_{22} & \cdots & \tilde{c}_{2N} \\ \tilde{c}_{31} & \tilde{c}_{32} & \cdots & \tilde{c}_{NN} \end{bmatrix}\right\}$$

- The difference between our validated and unvalidated model. with validated eigenvalues λ_m and un-validated eigenvalues $\tilde{c}_{mm}$,can be computed with the following Frobenius Norm or:

$$\left\|C_{XX} - \tilde{C}_{XX}\right\|_2^2 = \sum_{m=1}^{N} |\lambda_m - \tilde{c}_{mm}|^2 + \|C_{XX}\|_2{}^2 - \sum_{m=1}^{N} |\tilde{c}_{mm}|^2$$

- Once we have a distributional model we can use these models to form a likelihood function for testing whether our measured data conforms to our modeled distribution

$$L(x) = \log[\ \exp(1/2[(x - m_1)^T \boldsymbol{C_{xx1}}^{-1}(\boldsymbol{x} - \boldsymbol{m_1}) - (x - m_0)^T \boldsymbol{C_{xx0}}^{-1}(\boldsymbol{x} - \boldsymbol{m_0})])]$$

- From our log likelihood d is defined as

$$d^2 = (m_1 - m_0)^T \boldsymbol{C_{xx1}}^{-1}(m_1 - m_0)$$

- From this, we can configure our model for hypothesis testing. For the purposes of our system model this hypothesis testing determines whether our information system is within the specification of desired behavior H_0 or not H_1.

- The distribution of each of these hypotheses is as follows.

$$H_0\ L(x)\ N\left(\frac{-d^2}{2}, d^2\right)$$

$$H_1\ L(x)\ N(\frac{d^2}{2}, d^2)$$

- We can use this strategy to employ a hard threshold rule for detection such that

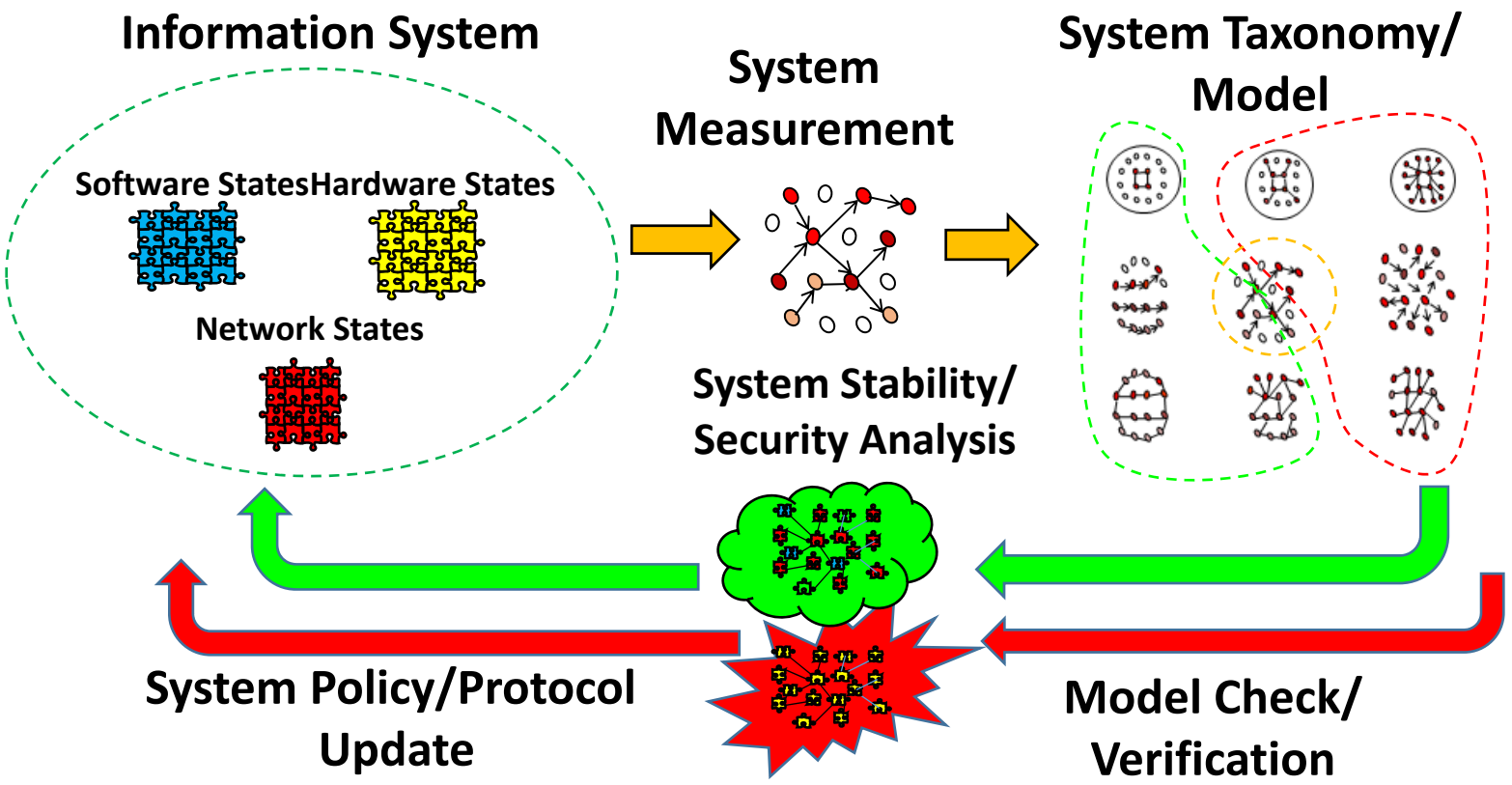
$$\mathrm{A}(x) = \begin{cases} 1\ L(x) \geq \eta \\ 0\ \ L(x) < \eta \end{cases}$$

- We can use this strategy to employ a hard threshold rule for detection such that

$$\mathrm{A}(x) = \begin{cases} 1\ L(x) \geq \eta \\ 0\ \ L(x) < \eta \end{cases}$$

Management

How do we close the loop at multiple architectural layers to assure mission performance and verify system policy/protocol is working?



Management

- To achieve a Bayes analytic risk function, we simplify the covariance model for our distribution using an L2 norm. Using the L2 norm gives us the largest eigenvalue of the Covariance matrix we have

$$\|C_{XX}\|_2 = \sigma^2{}_{max}$$

- Our new distributional metric becomes:

$$\hat{d}^2 = (m_1 - m_0)^T \sigma^{-2}{}_{max}(m_1 - m_0)$$

- To compute our Bayes risk, we compute a false-alarm probability, withthe false alarm probability as

$$\alpha = 1 - Q(z)$$

where $Q(z) = \int_{-\infty}^{z}(2\pi)^{-1}\, exp\{-x^2/2\}dx$

Management

- Our miss probability is $1-\beta$, where β

$$\beta = 1 - Q\left(z - \hat{d}\right)$$

- We now can state that η is our threshold derived from our desire to minimize the maximum risk of the worst case distribution.

$$z = \frac{\eta + \frac{\hat{d}^2}{2}}{\hat{d}}$$

- We now define our Bayes risk with

$$\Re(\alpha, \beta) = p_0 L_{01} \alpha + (1 - p_0) L_{10} (1 - \beta)$$

- with L_{01} as our respective loss when H1 is decided and and L_{10} our loss when H0 is decided when H1 is true and p_0 the probability of H0.

Management

Integrated Operation

- We either can manage the system by changing system parameters by minimizing the risk over the maximum eigenvalue of σ_{max} , or bychanging the threshold η to define new regions of performance.

$$\min_{\sigma_{max}} (\Re(\alpha, \beta)))$$

Risk Bounds

- Using this approach we can find an upper bound for the system risk to our distributed information system. The Chernoff Bound gives the average probability of error, P_e^* ,which can upper bound for our Bayes risk

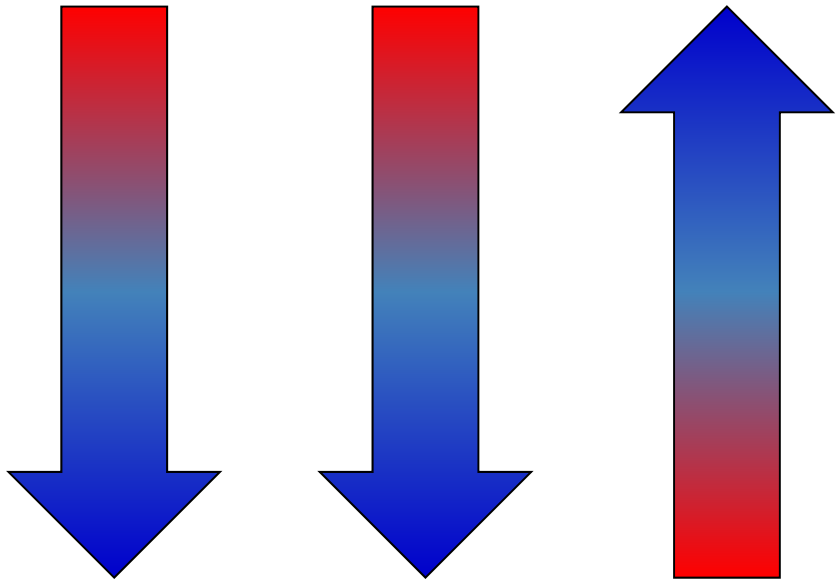
$$P_e^* \geq \exp(log \int f_{X|N}^{\alpha^*}(y|\Lambda = 1) f_{X|N}^{1-\alpha^*}(y|\Lambda = 0) dy)$$

$$\alpha^* = min_{0 \leq \alpha \leq 1} \int f_{X|\mathrm{N}}^{\alpha^*}(y|1) f_{X|N}^{1-\alpha^*}(y|0)$$

Metrics of Performance

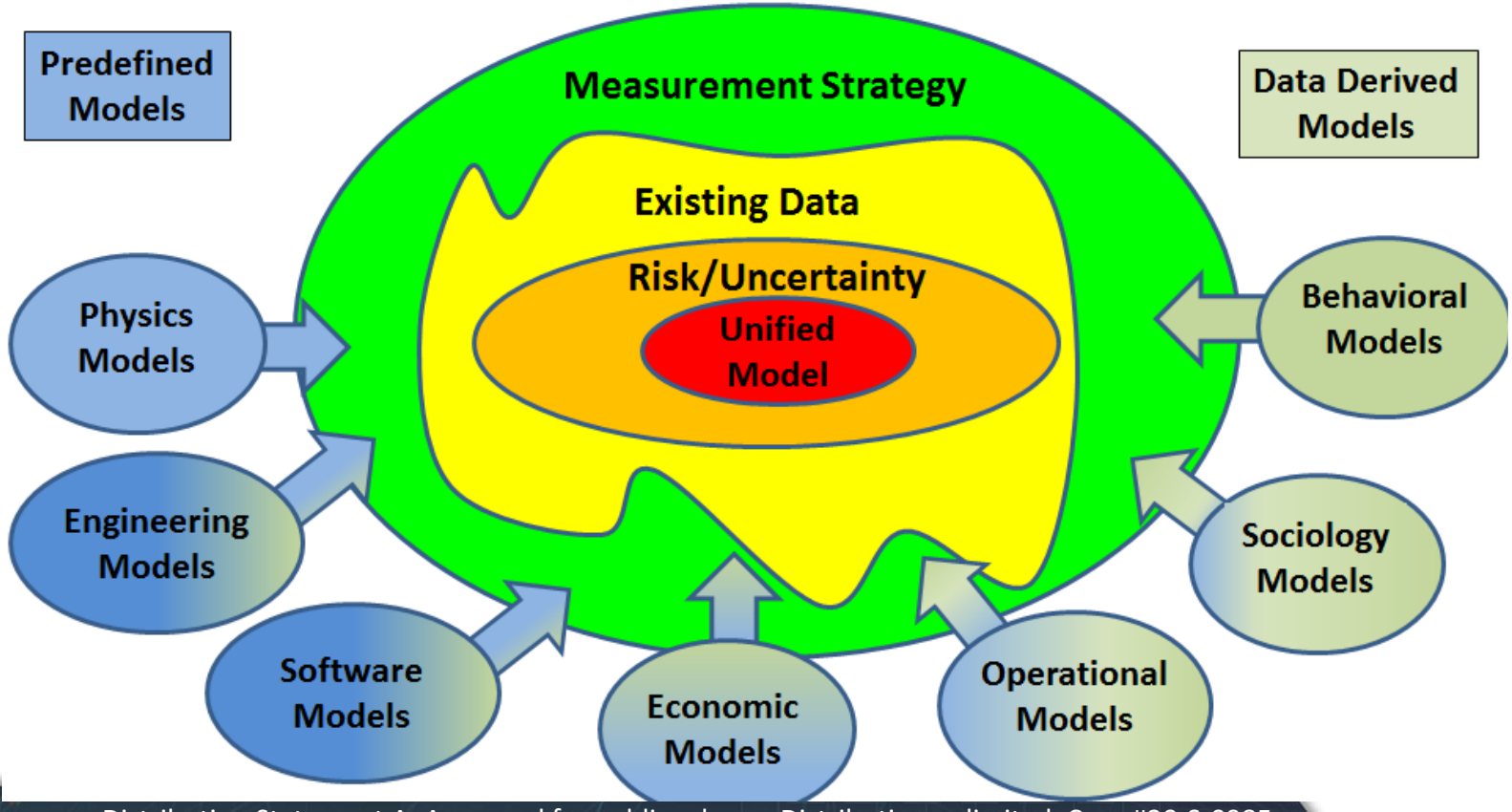
Metrics of performance allow timelines, tracking, and mission performance to be rigorously assessed by analyst/commander in real time.

Example Metrics	Desired Outcome
Timeline Reduction	Analysis Time
Rigorous Mission Threat/Risk Assessment	Mission Risk
Rigorous Data Product Confidence Analysis	Data Product Confidence



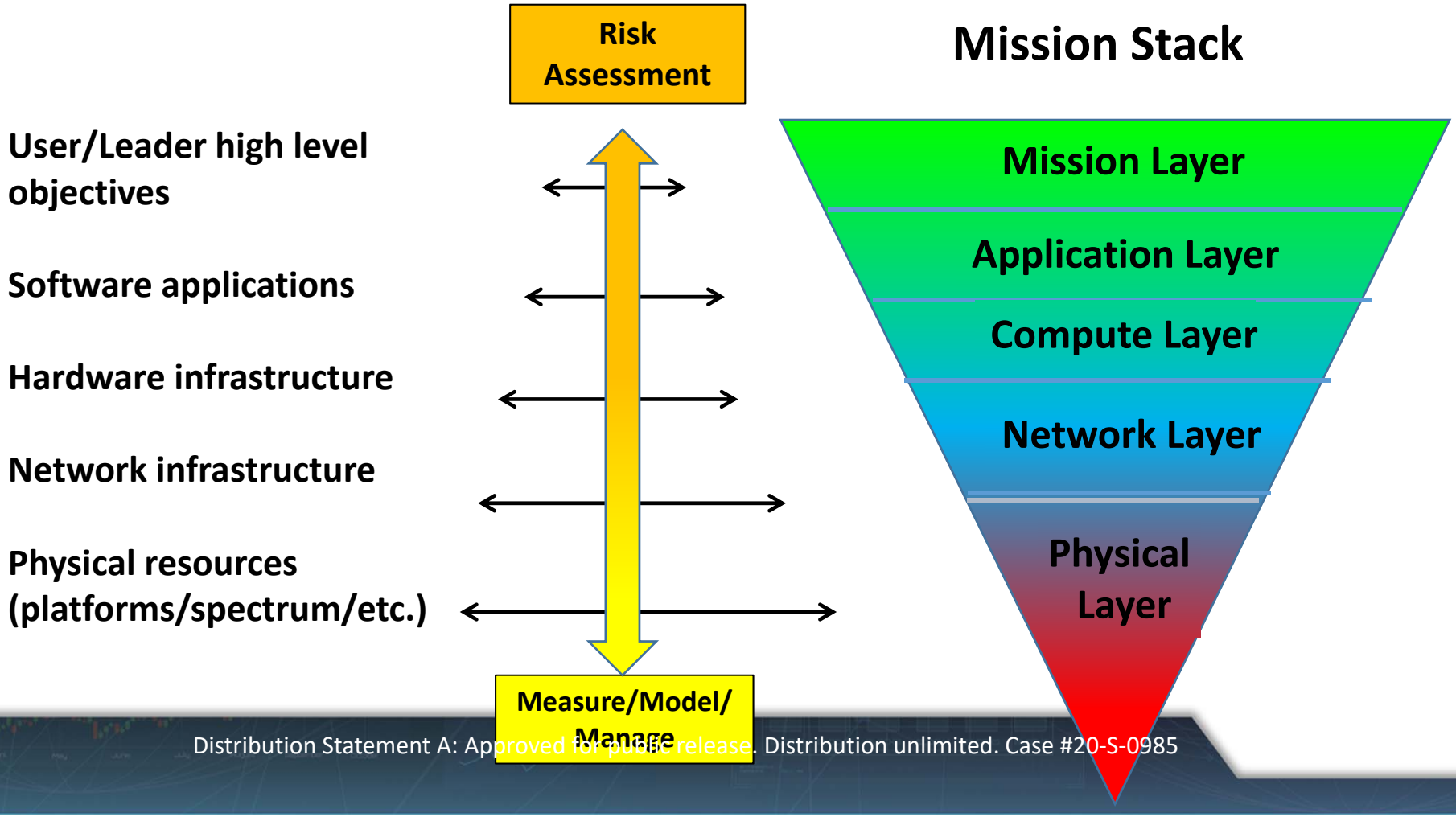
Risk Analysis and Modeling

Unified methods for data modeling require a rigorous risk assessment in order to assure commanders, analysts, and system operators of performance.



Mission Stack

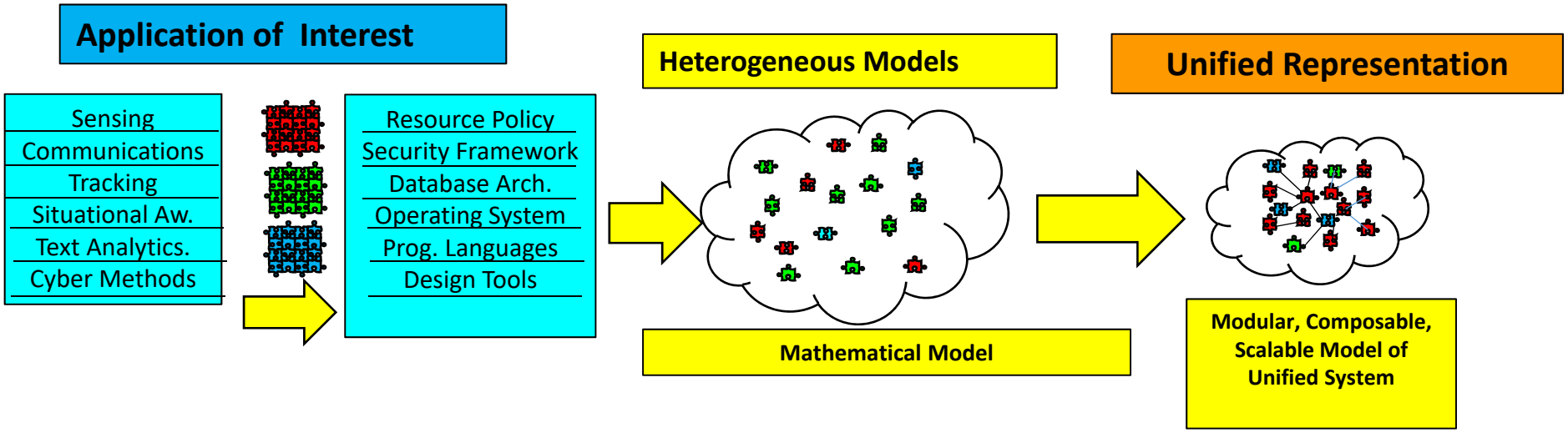
Measurement, modeling, and management of mission stack must have rigorous performance and risk metrics associated with them



Application Layer

The mission layer may be made up of multiple applications such as sensing, communication, tracking, situational awareness, command and control, etc.

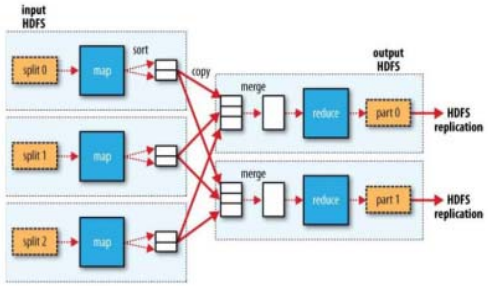
-These methods must be integrated with one unified representation for validation and verification.



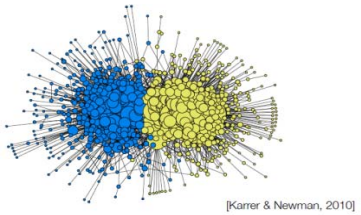
Compute Layer

Current computational infrastructures (cloud resources) are currently highly distributed and resource allocation is static. Making this process more dynamic will create resilient system performance.

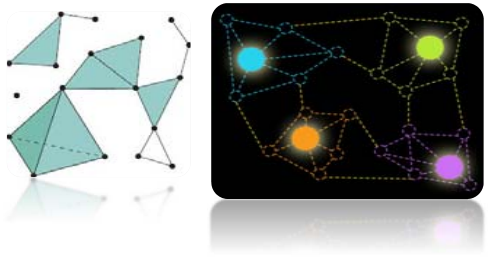
Critical Apps on MAP-Reduce Cloud Computing Engine



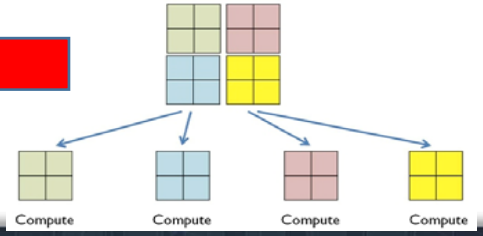
Measurement Based Graph Analytics



System Performance Verification

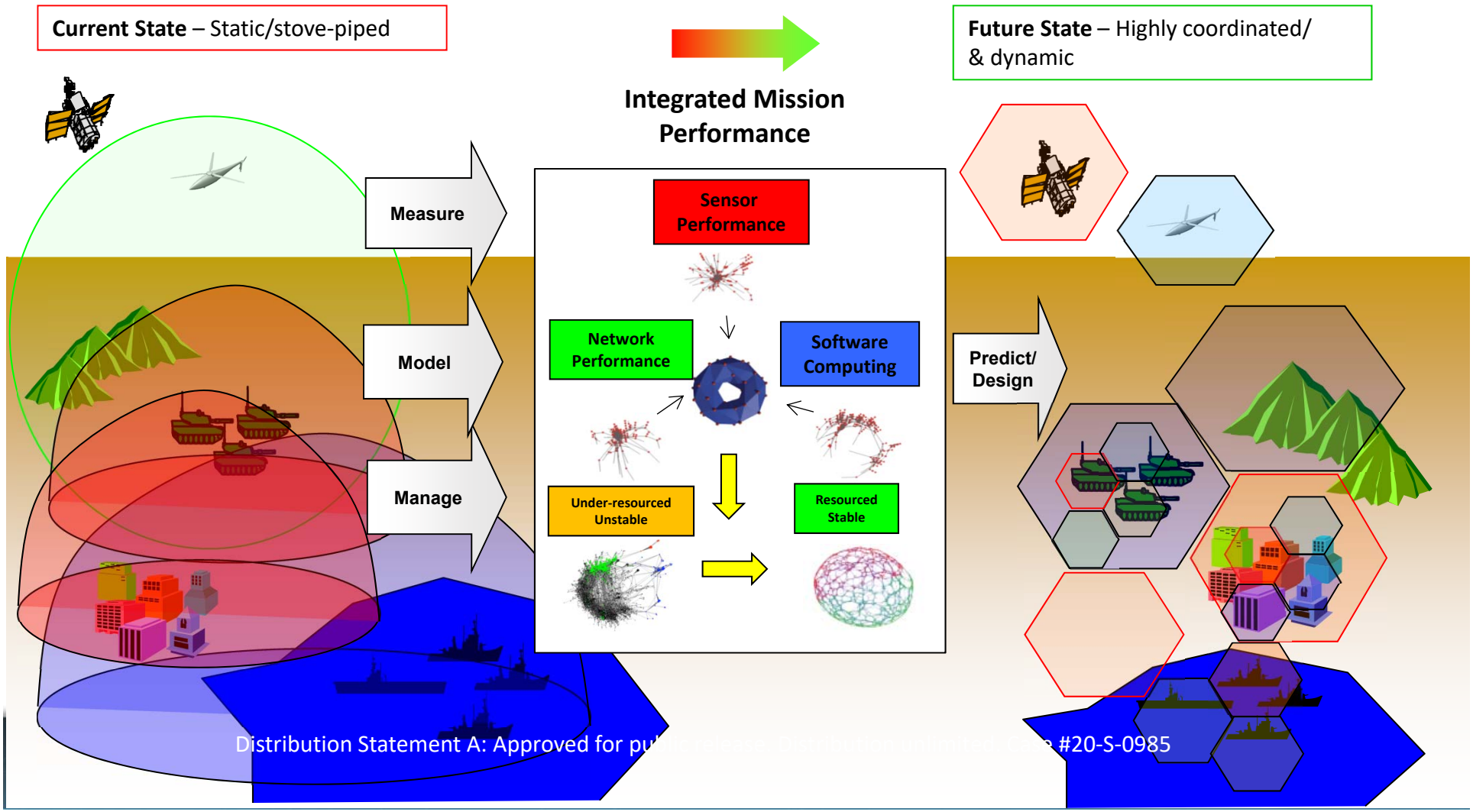


Computed System State Representation



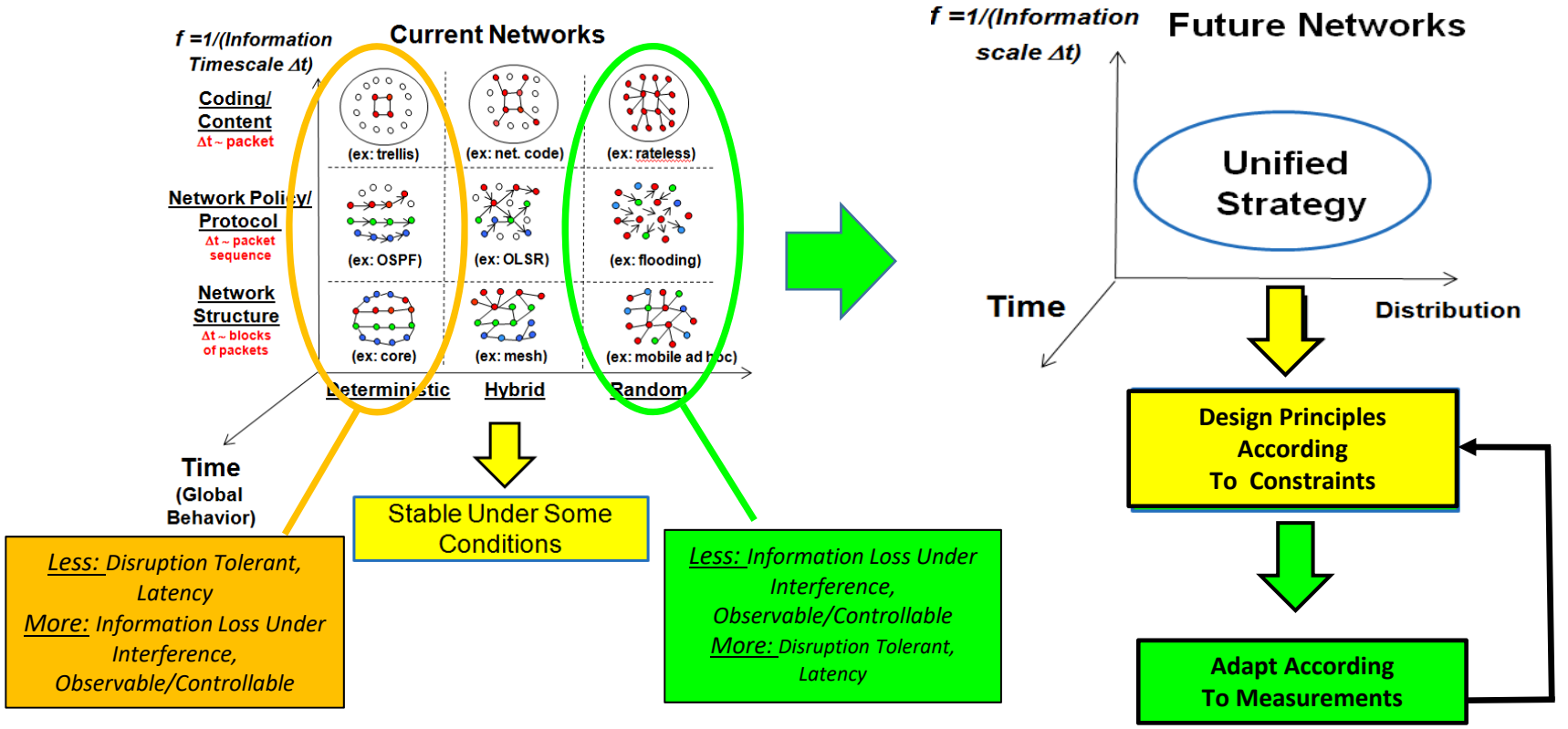
Physical Layer

Pressure on spectrum is changing the static and highly segregated assumptions about physical layer performance.



Network Layer

Advances such as software defined networks are changing stove piped network management to a heterogeneous management problem which requires dynamic assessment



Unified Operation

Measure and verify information system properties among various system constraints