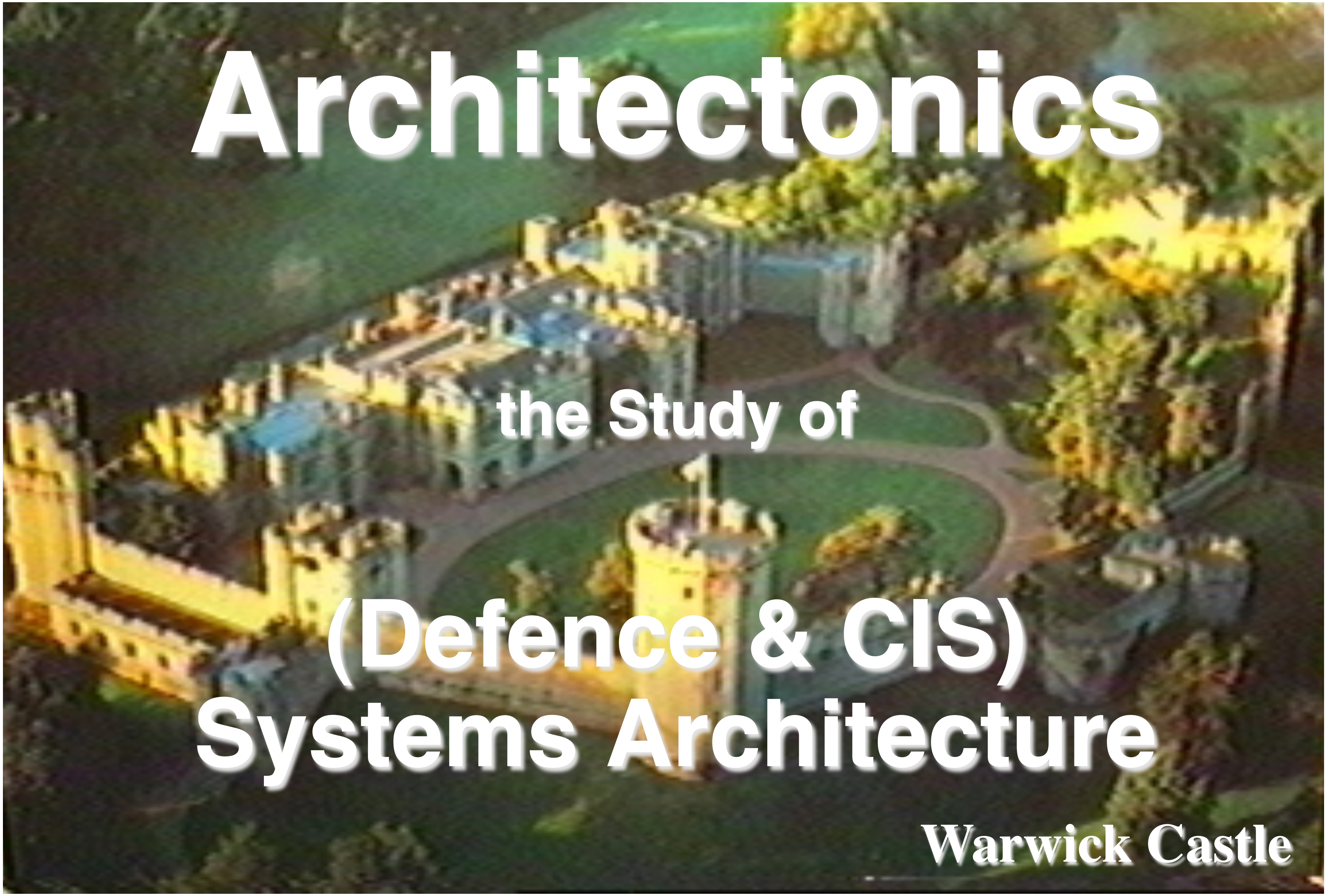
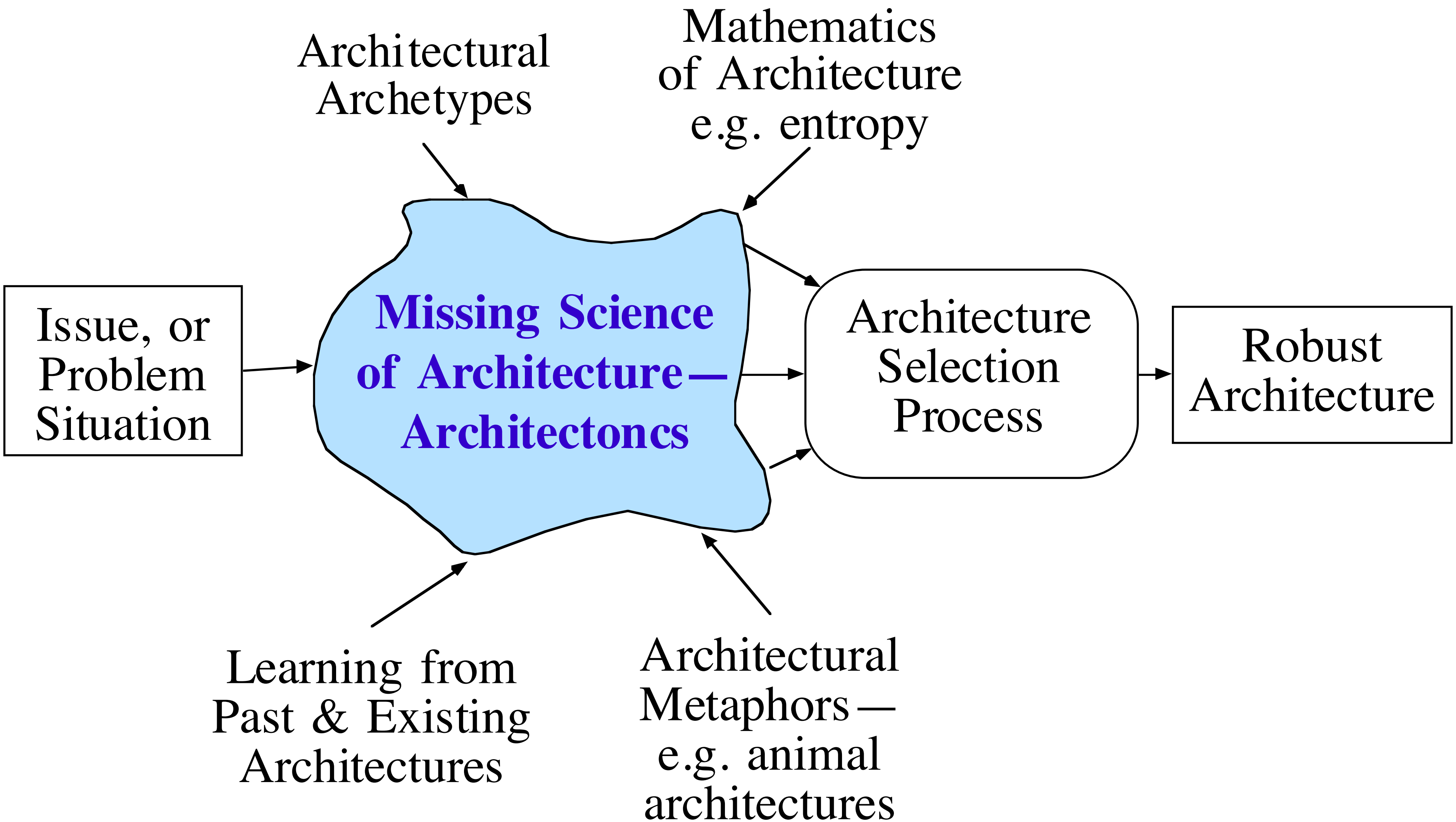

Architectonics
the Study of
(Defence & CIS)
Systems Architecture
Warwick Castle

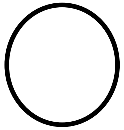
Systems Architecture

- At one and the same time:—
  - Least understood feature of Command & Control Systems, but…
  - …most significant feature:—
    - » systemic—affects everything
    - » affects decision speed
    - » connects decision-makers to information
    - » connects decision-makers to forces, i.e. enables control
    - » tolerates damage, i.e. reconfigures
    - » self-heals, i.e. repairs itself (with/without human-help)
- Different Missions dictate different architectures to enable & support different levels of Performance, Survivability, etc. Yet…
- There is no science relating task to corresponding “best” architecture

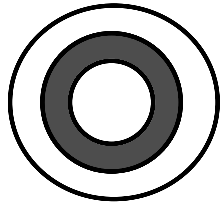
Architectonics—the Missing Science



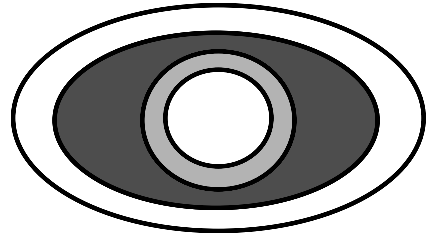
Animal Architectures



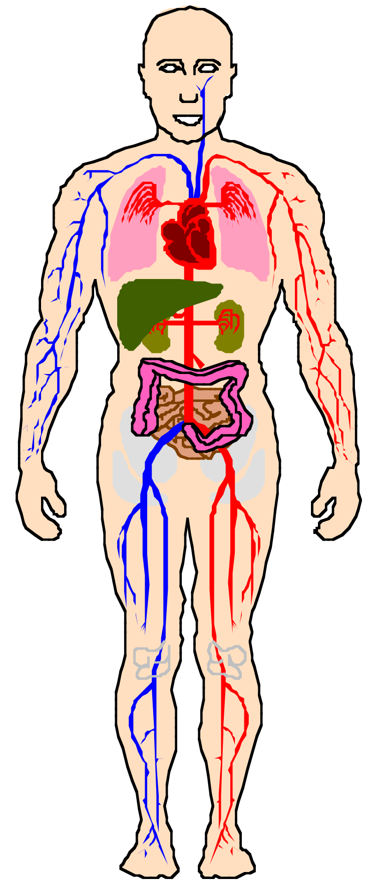
unicellular
e.g. protozoan

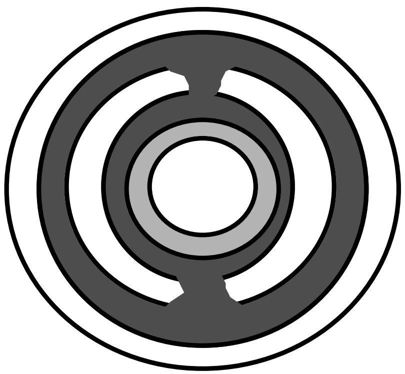


two cell layers
e.g. sponge



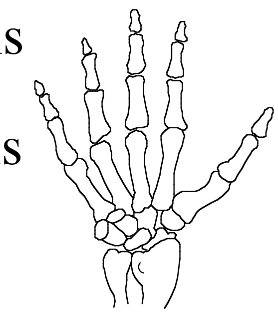
three cell layers





three cell layers
and body cavity

- hydrostatic skeletons
- stiff skeletons
- exoskeletons
- endoskeletons

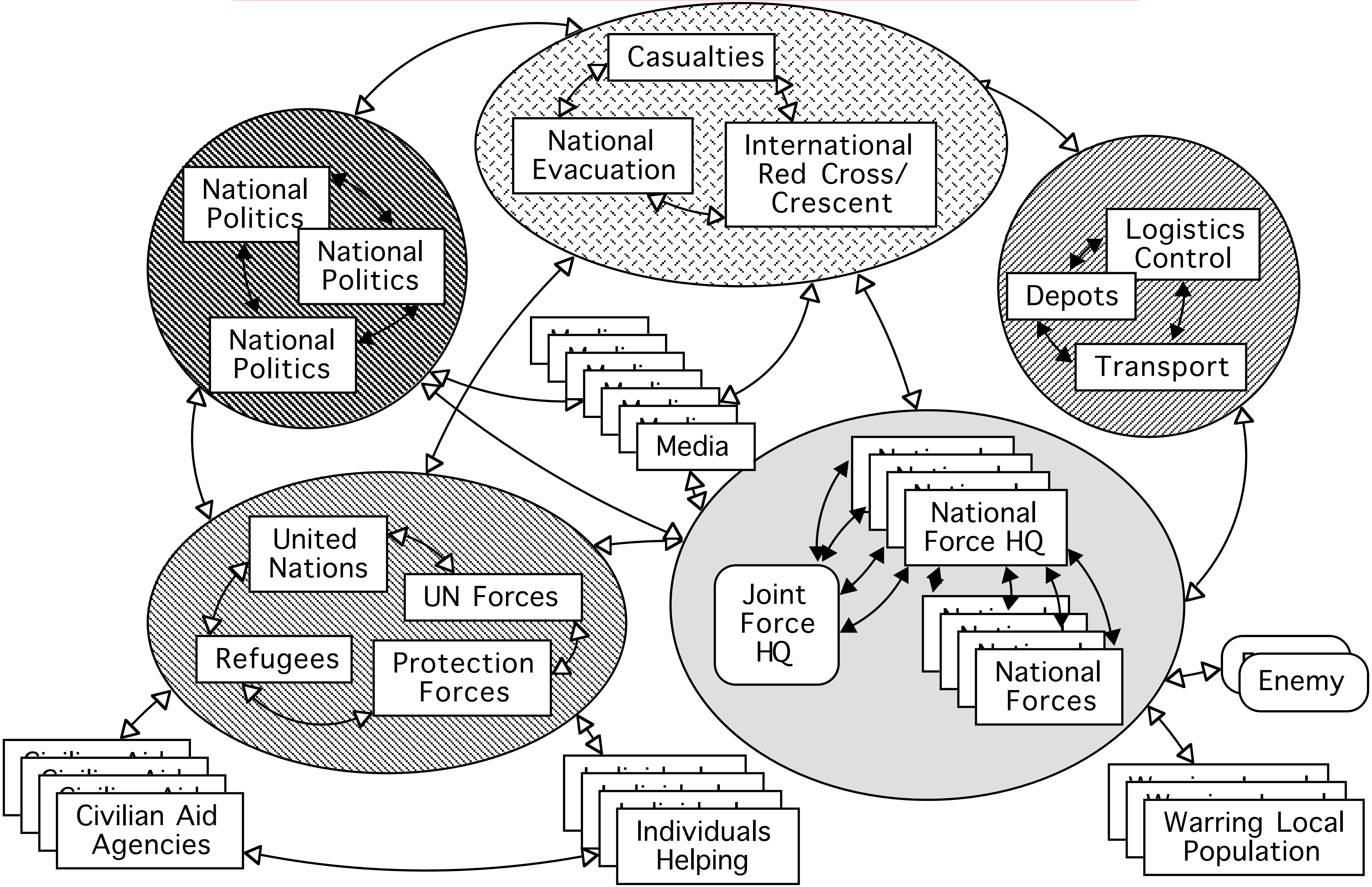


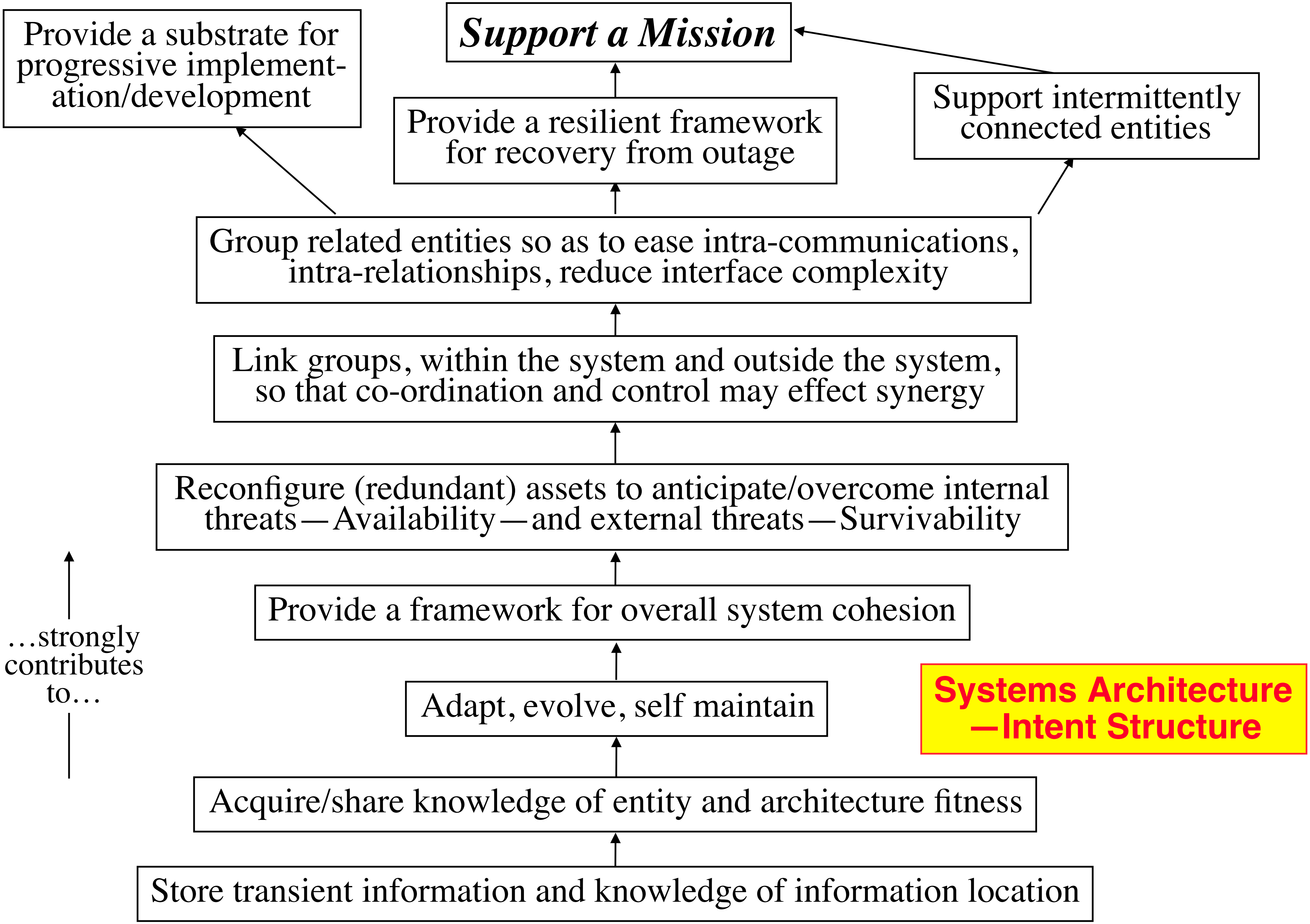
Key deductions include:—minimum variety for viability, importance of waste disposal, non-scalability, physical protects vulnerable communications, organs

Alien Animal Architecture?

- The microbe from Mars(?)
- Order midst disorder

Typical Interacting Command Architectures




Support a Mission
Provide a substrate for progressive implementation/development
Provide a resilient framework for recovery from outage
Support intermittently connected entities
Group related entities so as to ease intra-communications, intra-relationships, reduce interface complexity
Link groups, within the system and outside the system, so that co-ordination and control may effect synergy
Reconfigure (redundant) assets to anticipate/overcome internal threats—Availability—and external threats—Survivability
Provide a framework for overall system cohesion
Adapt, evolve, self maintain
Acquire/share knowledge of entity and architecture fitness
Store transient information and knowledge of information location
…strongly contributes to…
Systems Architecture —Intent Structure

Kinds of Architecture

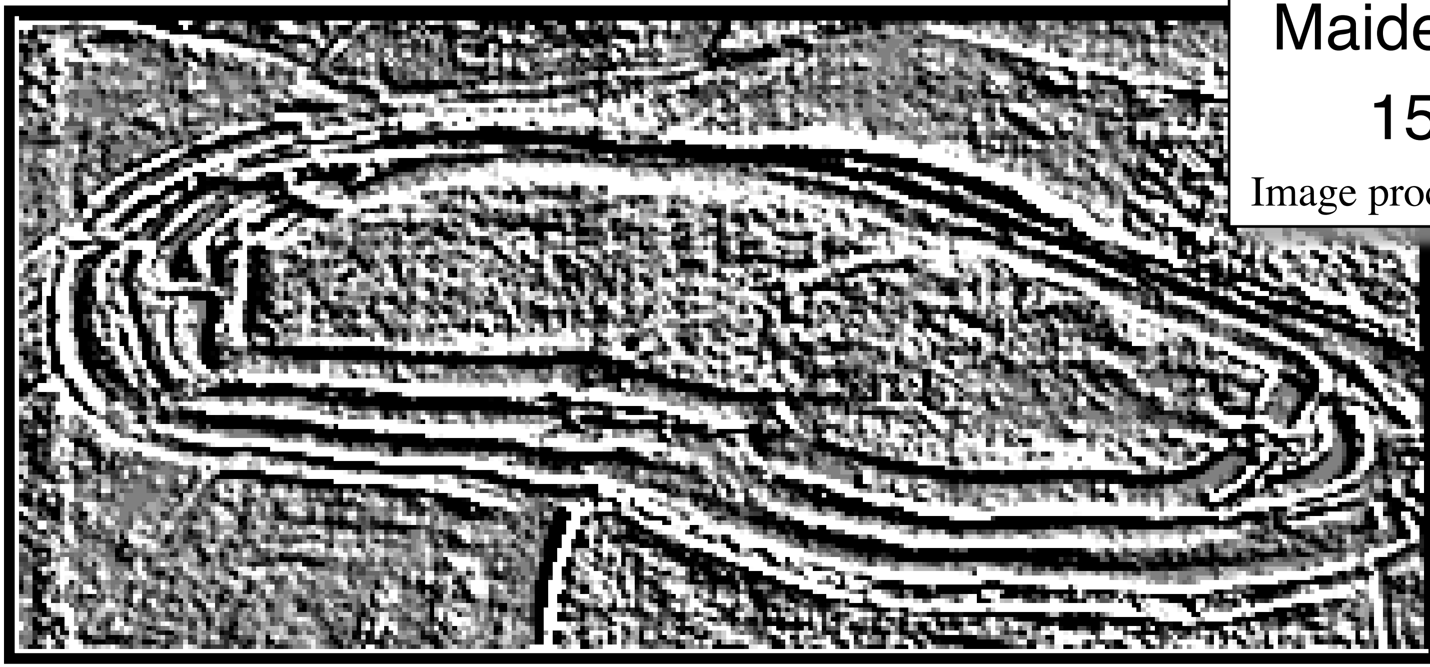
- Structure offers two main archetypes: —
  - ***Layered architectures***,
    - » enabling or resisting passage through successive layers.
    - » process-oriented manufacturing, communications, defence, security, trees and plants, Sun, alimentary canal…
  - ***Clustered architectures***
    - » grouping reduces component interaction energy.
    - » human organization, circuit board and microcircuit design, biological “designs”, book topics, warehouses, ethnic restaurants, libraries…


Bodiam
Castle
Layered
Architectures

Learning from History

- Designing **optimum** systems *ab initio* “difficult to impossible”
- **Best systems evolve**. Takes:—
  - harsh, varied, testing threat environment, real failures, trial and error
  - time
- Many present-day systems **never tested in anger**—will designs prove effective…?
- One approach—**learn from history**, but…
  - you have to read history carefully and translate the lessons into present *and future*

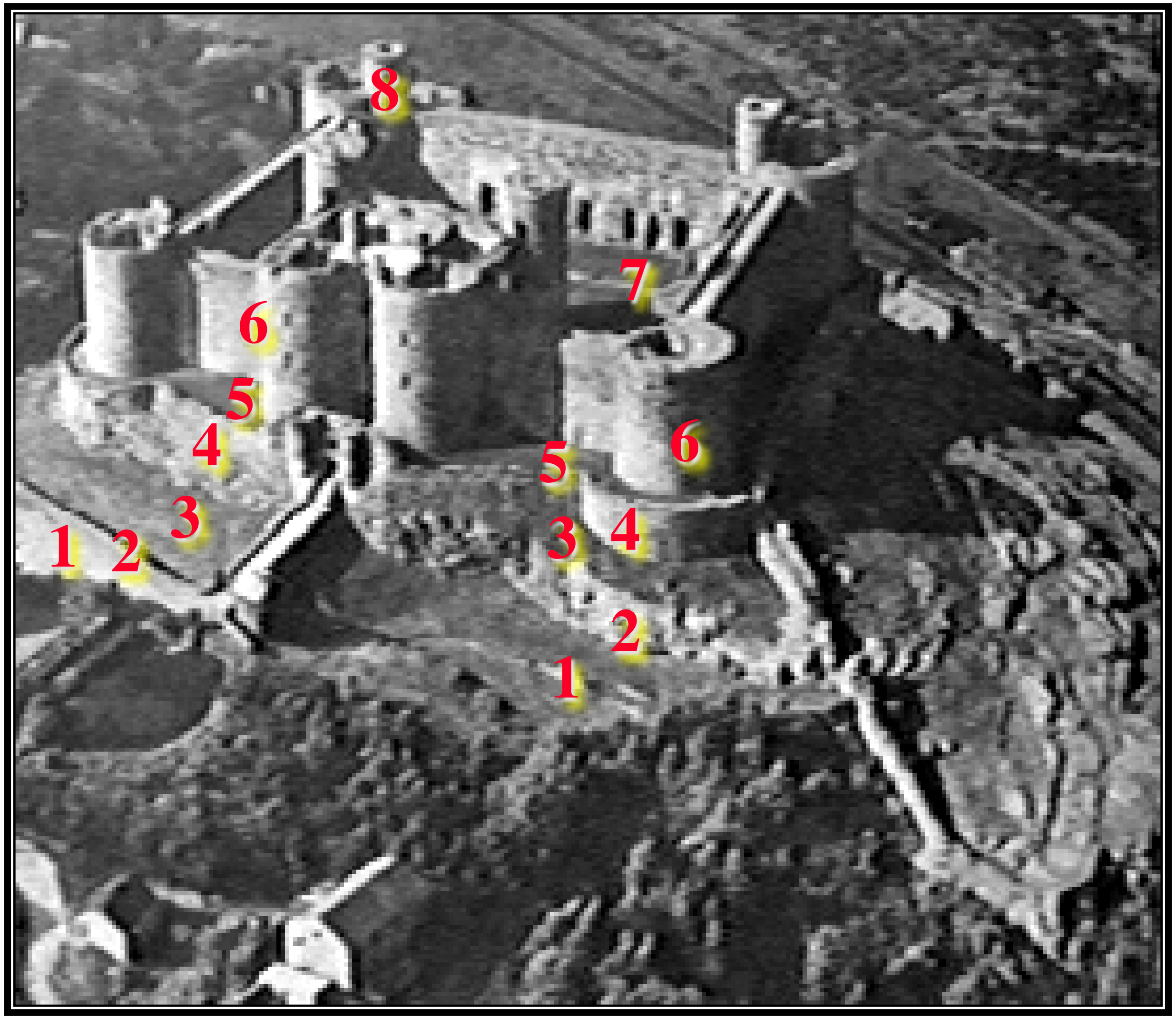
…So, How Many Layers?



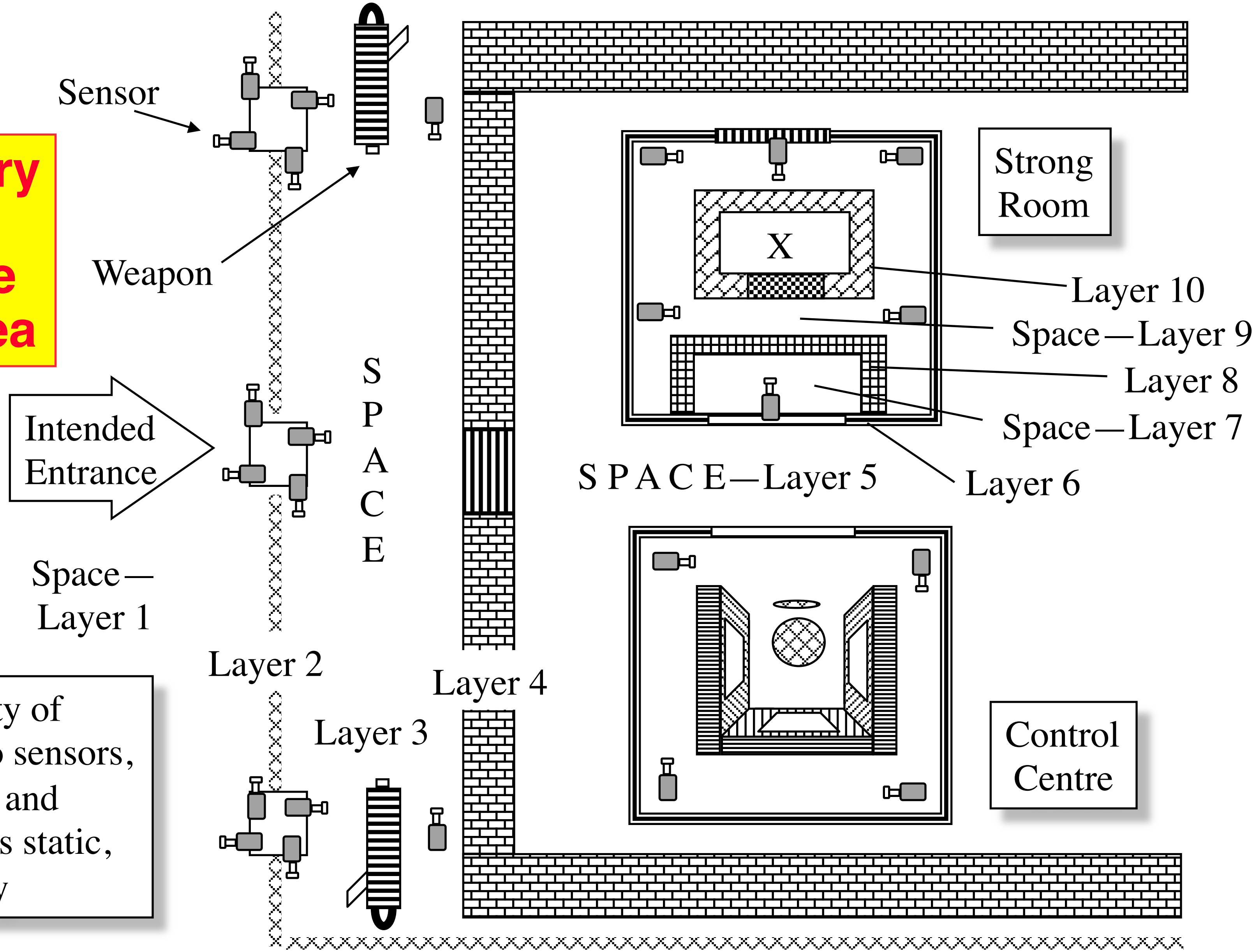
Maiden Castle, Dorset—
1500BC—450AD
Image processed to enhance layer visibility

- Counting earth ramparts, 4/5 layers of defence
- Additional earthworks guard entrances—always a weak point
- Counting ditches between ramparts too, **≥ 7**
- **≥ 7** *appears repeatedly in all kinds of architectures…*

Layered Defence— Harlech Castle *1277-1330*

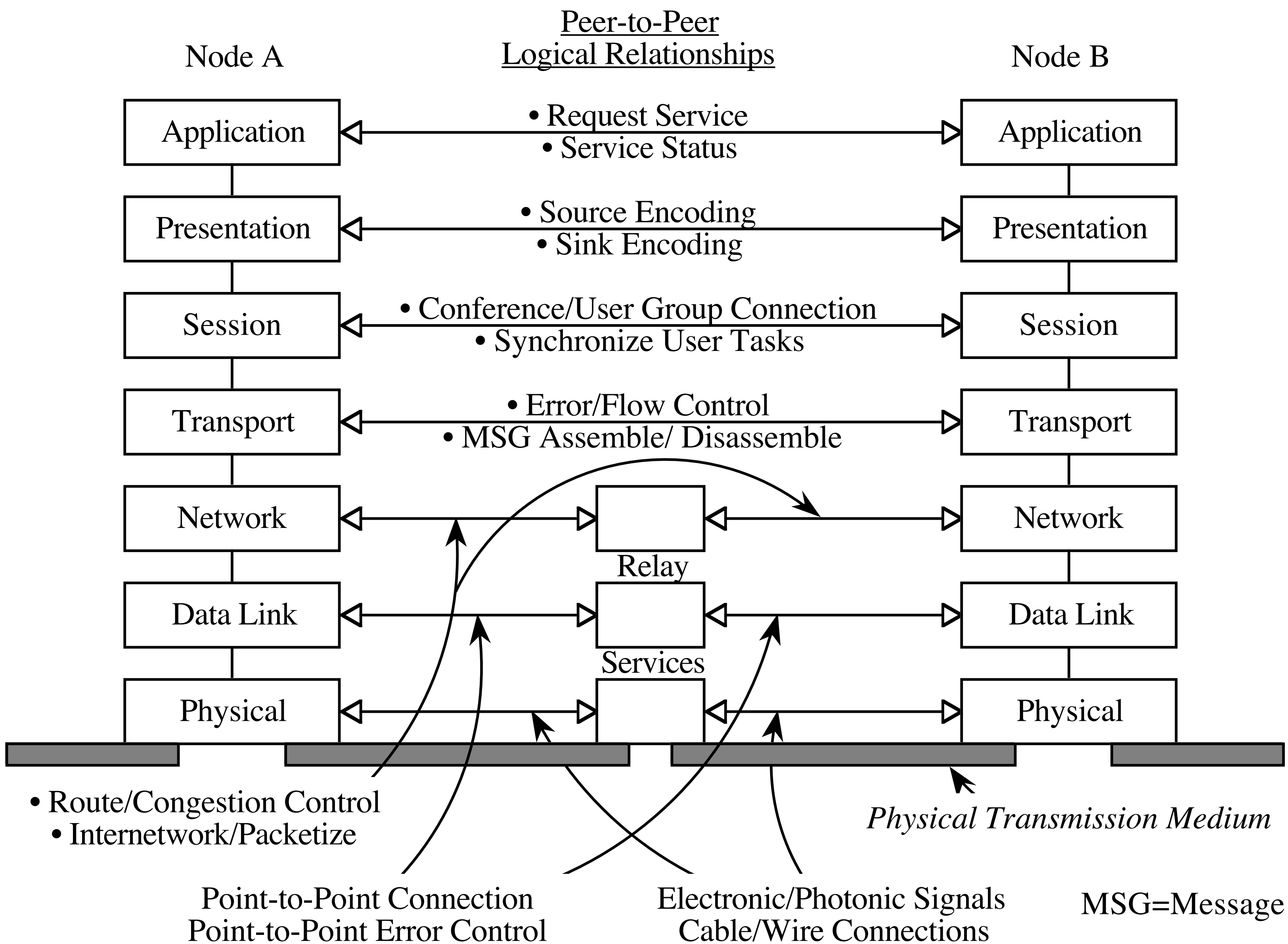


Contemporary Layered Architecture —Secure area



Note vulnerability of overall system to sensors, communications and control. Design is static, lacks redundancy

ISO Open Systems (7 layer) Interconnection



Simple Mathematics of Multi-Layered Defence

$$P = 1 - (1 - p)^N$$

where…

p…is the neutralization probability per layer

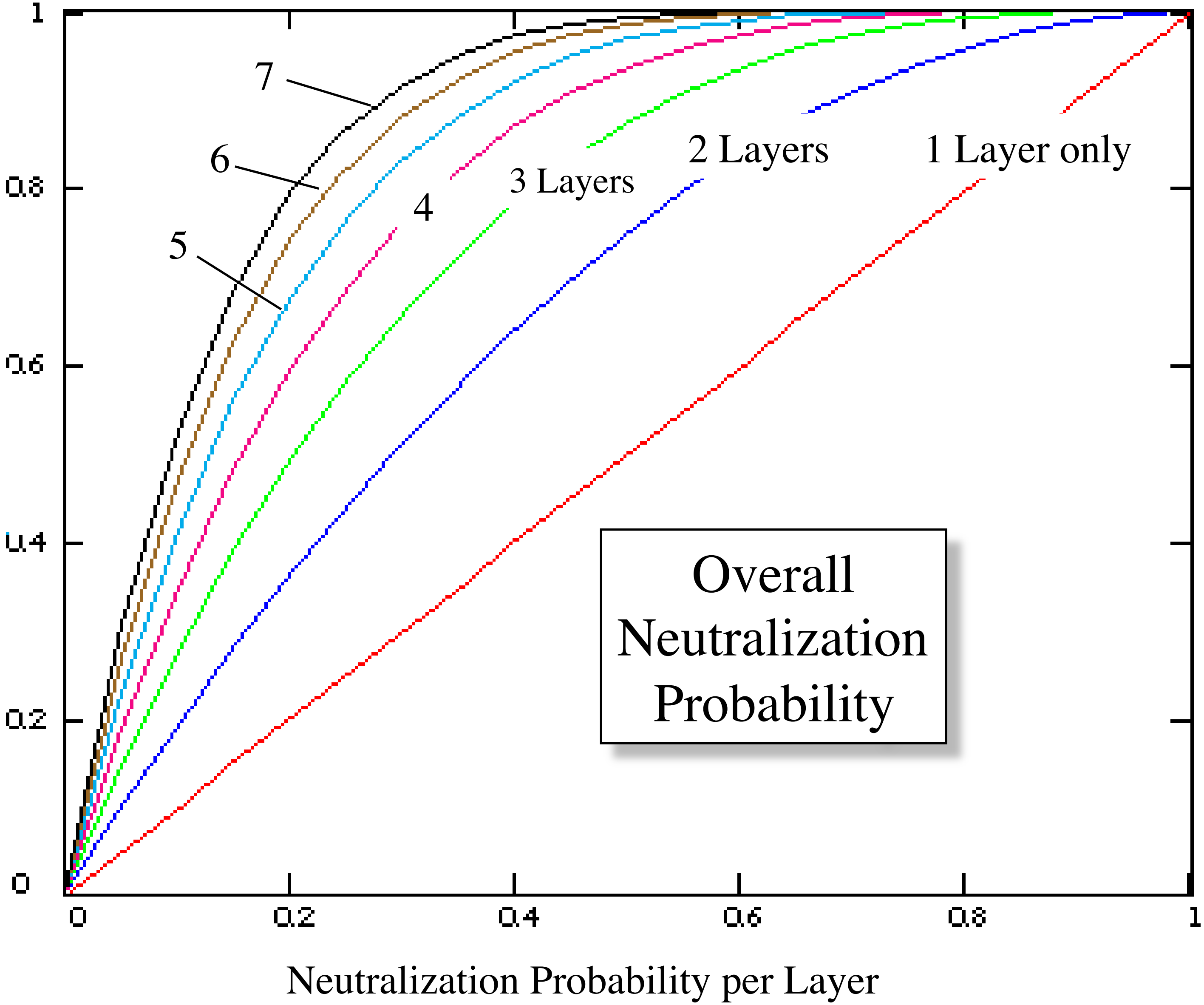
N…is the number of layers

P…is the expected overall Neutralization

N.B. Assumes all layers have equal p

(1-p) is the *leakage* probability per layer

Simple Multi-Layer Maths

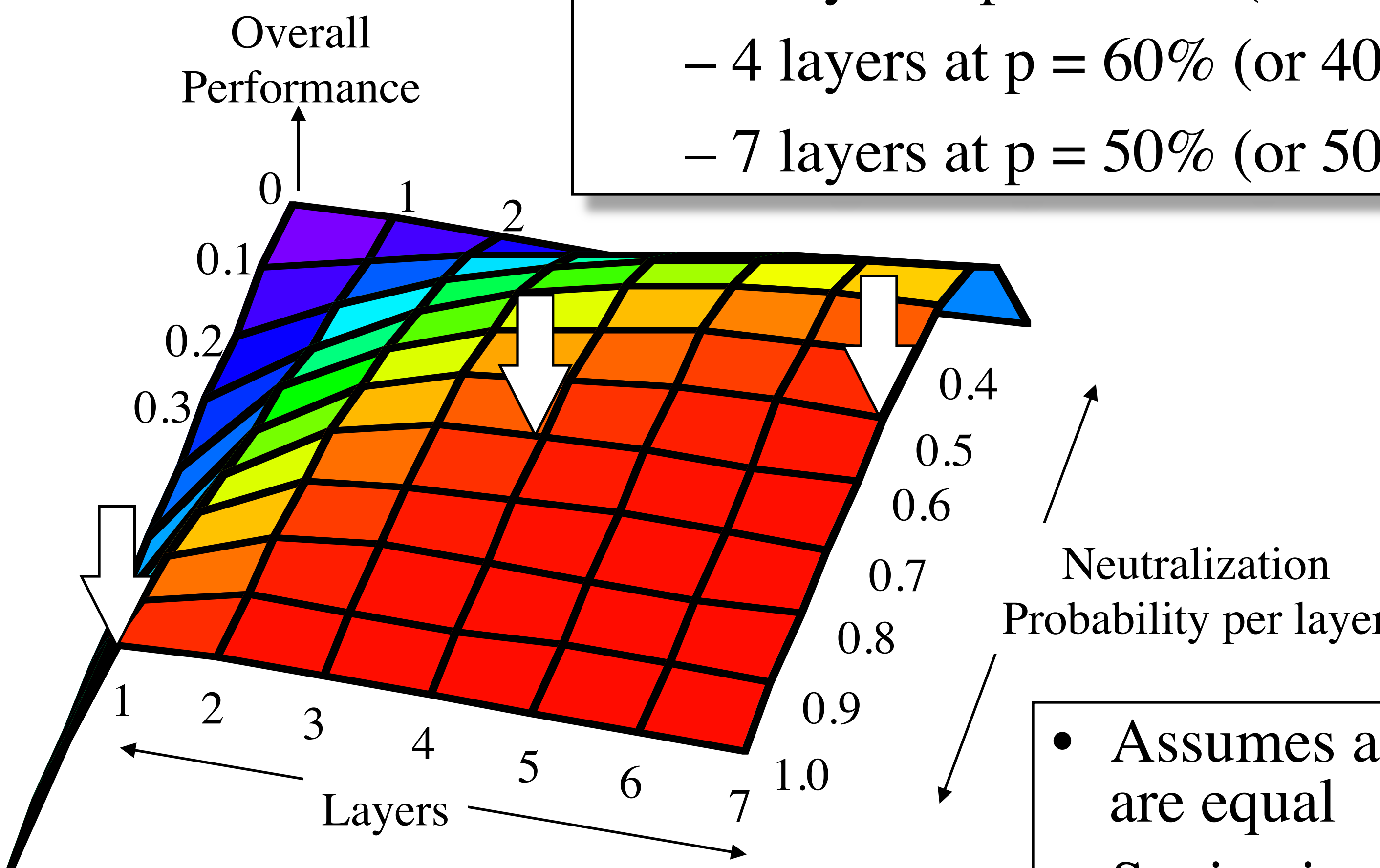


- One Layer alone vulnerable
- Must give v. high protection
- Difference between 6 and 7 layers v. small

N.B. Mathematics assumes all layers equal

Layered Defence—Performance Optimization?

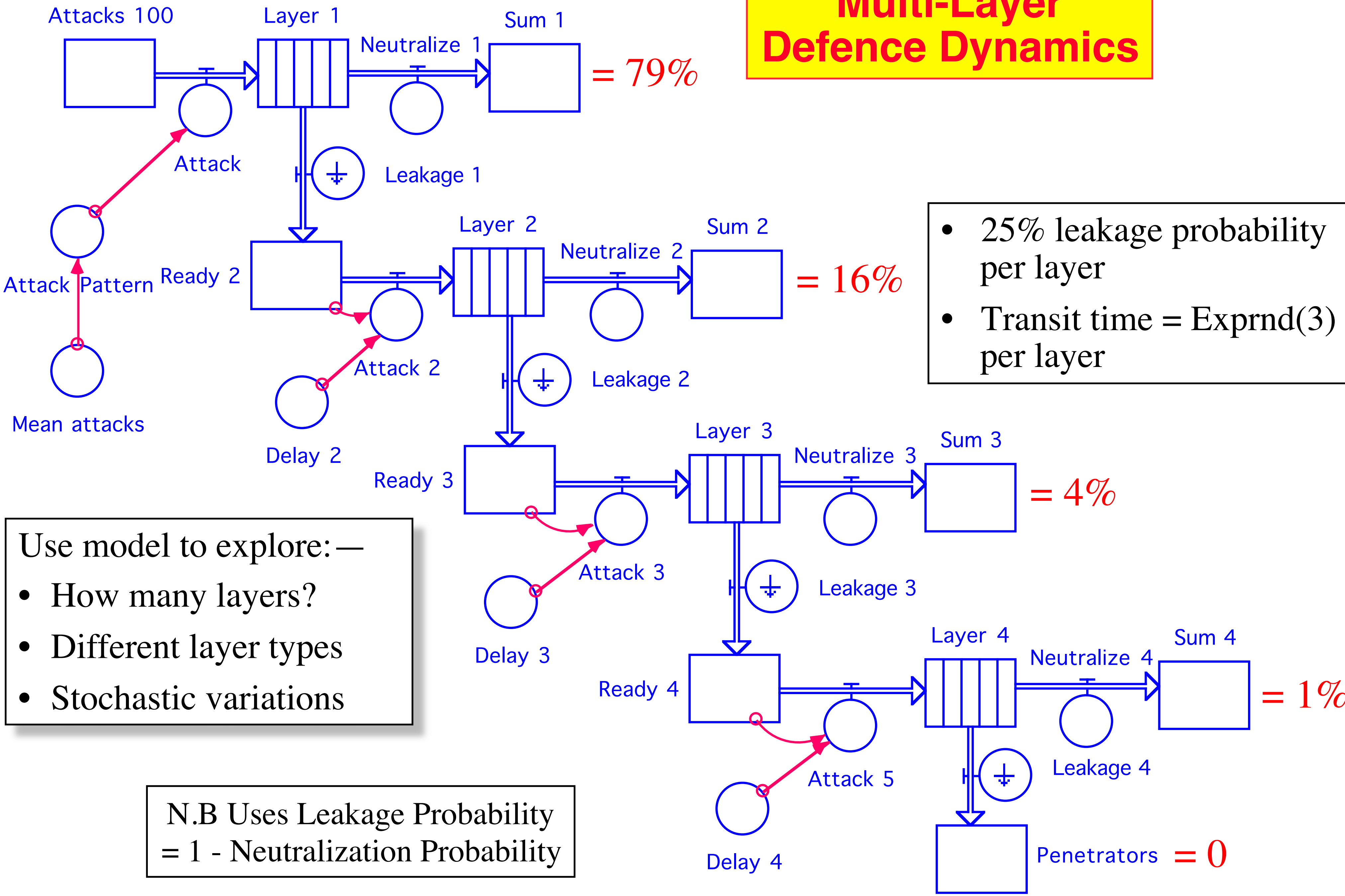
- Optimum overall performance:—
  - 1 layer at p = 100% (or 0% leakage)
  - 4 layers at p = 60% (or 40% leakage)
  - 7 layers at p = 50% (or 50% leakage)



- Assumes all layers are equal
- Static viewpoint

Multi-Layer Defence Dynamics

- 25% leakage probability per layer
- Transit time = Exprnd(3) per layer

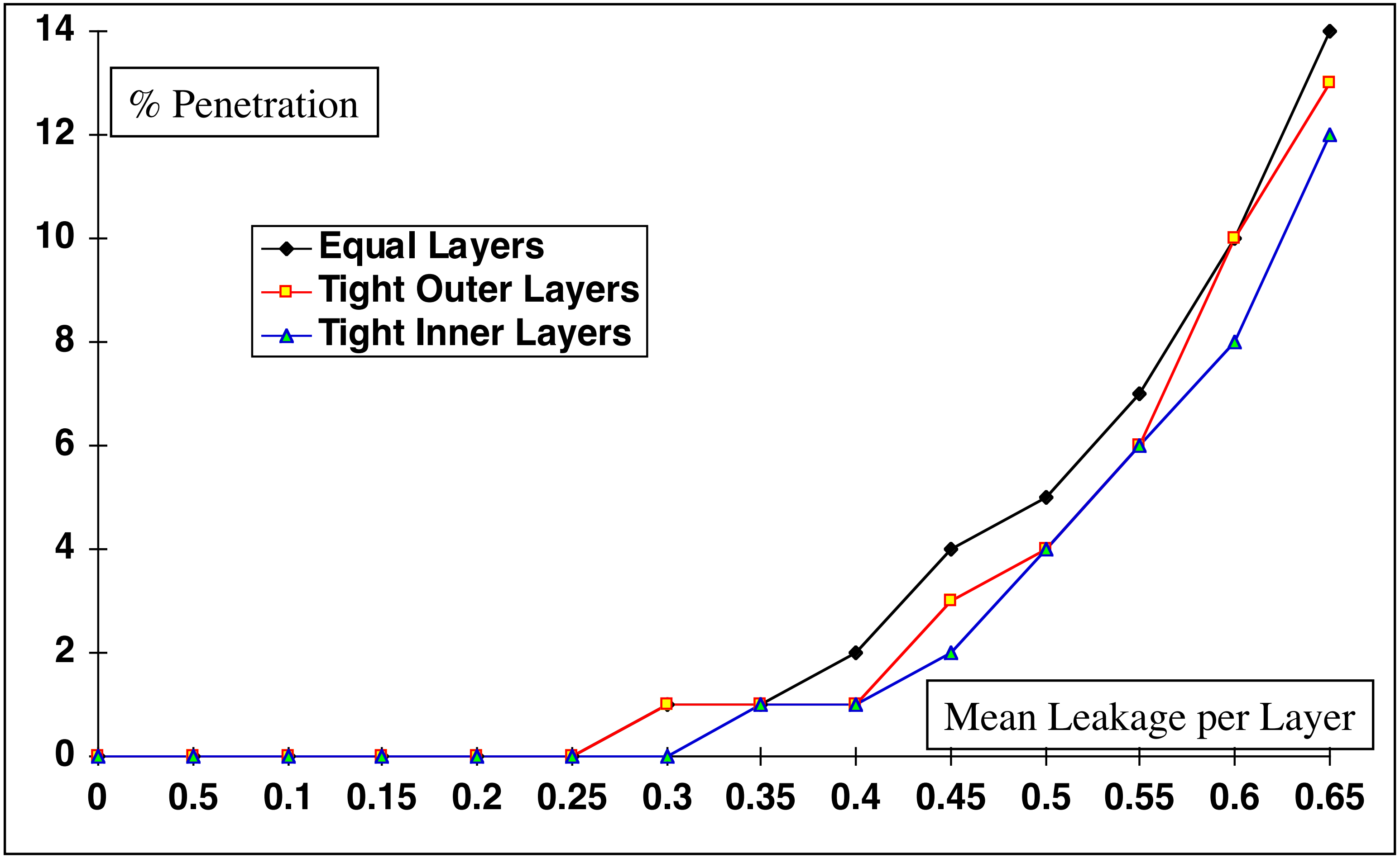


Use model to explore:—

- How many layers?
- Different layer types
- Stochastic variations

N.B Uses Leakage Probability = 1 - Neutralization Probability

Variation in Four-Layer Performance



Simulation of 4-layer defence-in-depth

Equal Layers—layer has same leakage

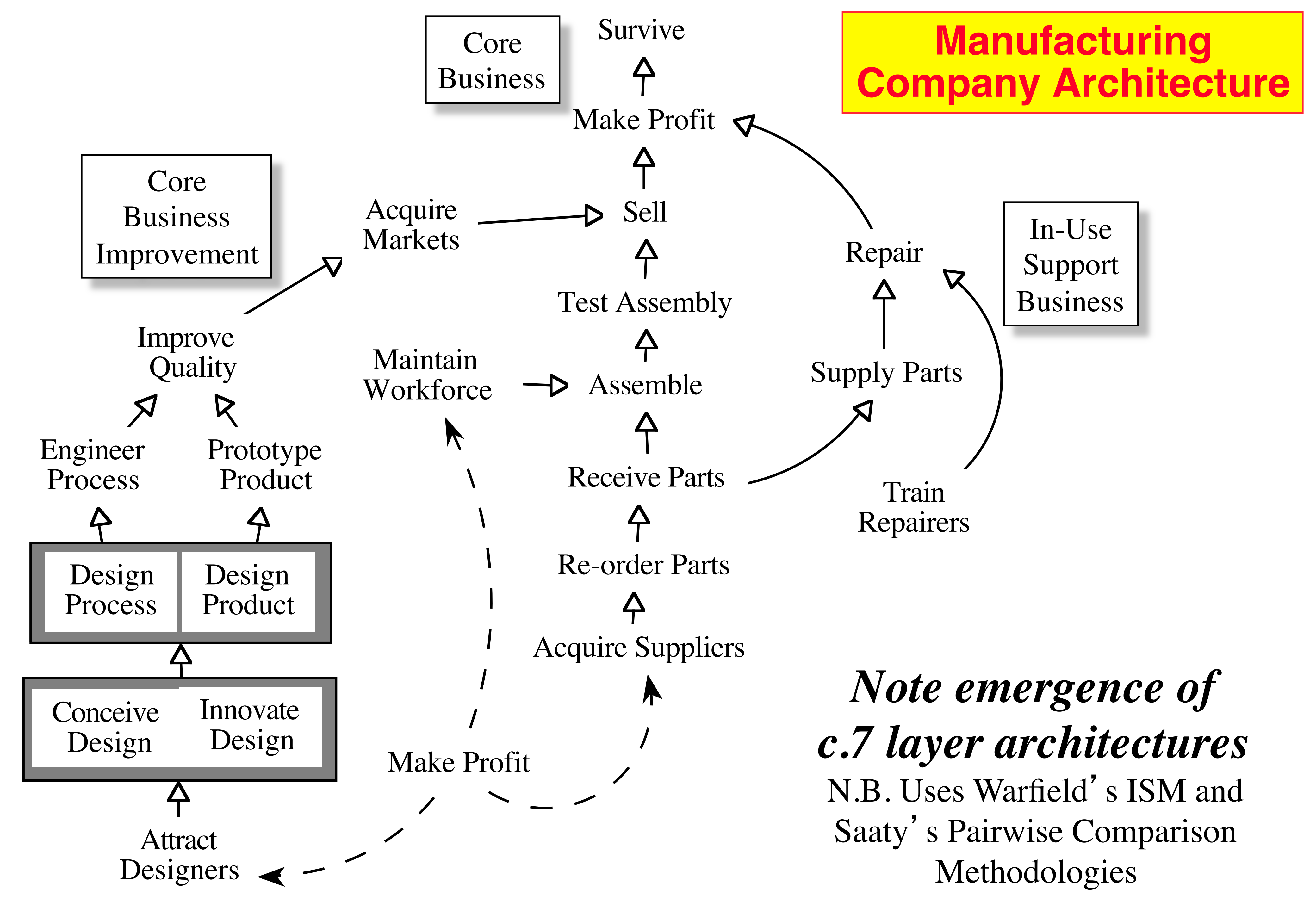
Tight Outer, Loose Inner Layers—leakage lower on outer layers, higher on inner layers

Tight Inner , Loose Outer Layers—leakage higher on outer layers, lower on inner layers

- Against intuition, best overall performance (lowest % penetration) corresponds to tight *inner* layers, i.e. loose *outer* layers
- Arises because of more even workload share between layers

Developing Architecture from Task, Activity and Process

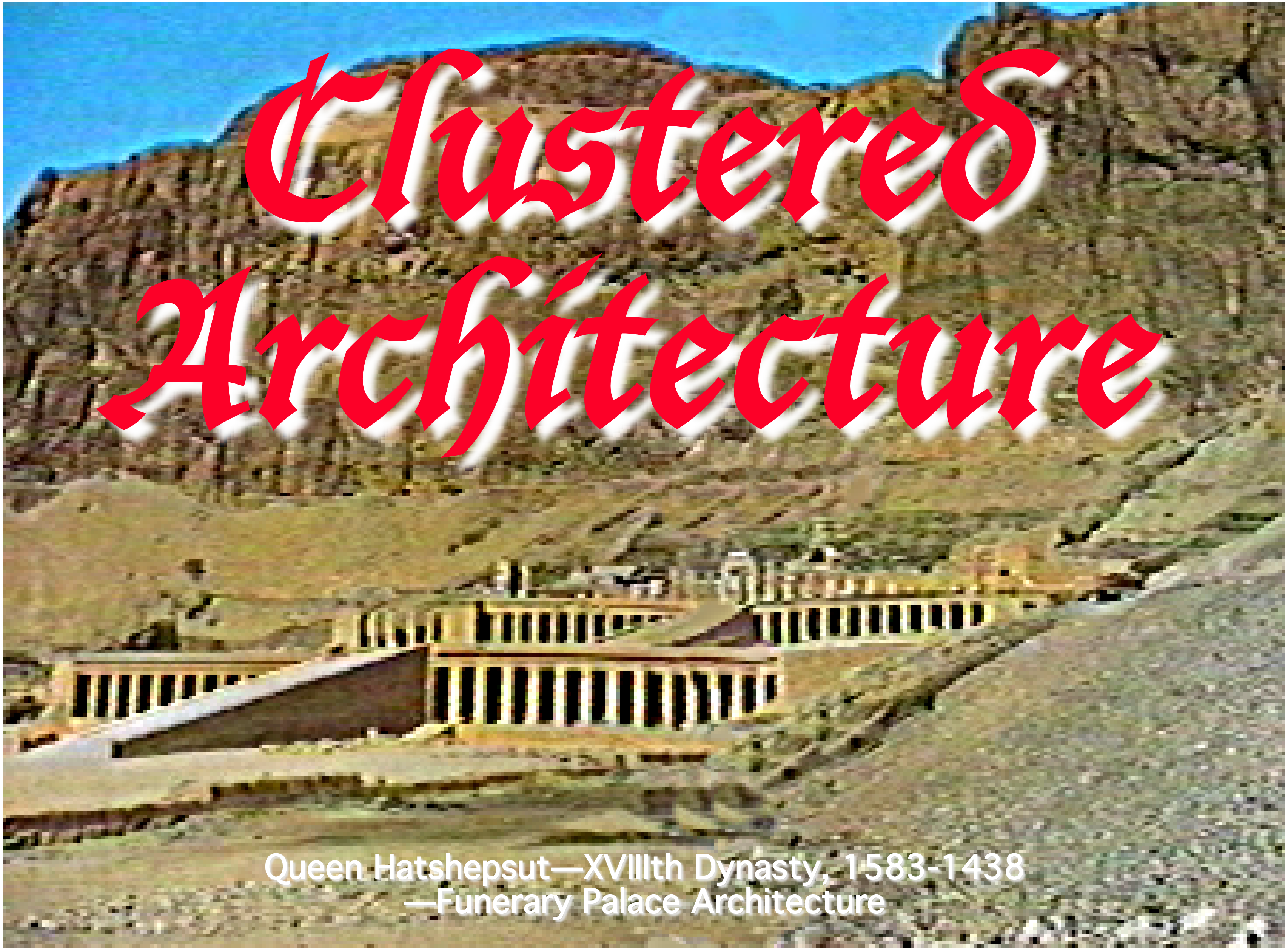
- **Step 1.** Identify separate Tasks, Activities, Processes
  - e.g. • Acquire Suppliers • Order Parts • Receive Parts • Assemble • Test Assembly • Sell • Make Profit • Survive • Repair • Supply Parts • Train repairers • Innovate Design • Attract Designers • Improve Quality • Conceive • Design New Product • Design New Process • Prototype Product • Engineer Process • Acquire Markets • Maintain workforce
- **Step 2**. Establish relationships between every task/ activities/process on a pairwise basis (SAATY)
- **Step 3**. Develop architecture using layers and clusters emerging from relationship matrix (Warfield's ISM)



Manufacturing Company Architecture
Core Business
Survive
Make Profit
Sell
Acquire Markets
Test Assembly
Assemble
Maintain Workforce
Receive Parts
Re-order Parts
Acquire Suppliers
Make Profit
Core Business Improvement
Improve Quality
Engineer Process
Prototype Product
Design Process
Design Product
Conceive Design
Innovate Design
Attract Designers
In-Use Support Business
Repair
Supply Parts
Train Repairers
Note emergence of c.7 layer architectures
N.B. Uses Warfield's ISM and Saaty's Pairwise Comparison Methodologies

Layered Architecture—Summary

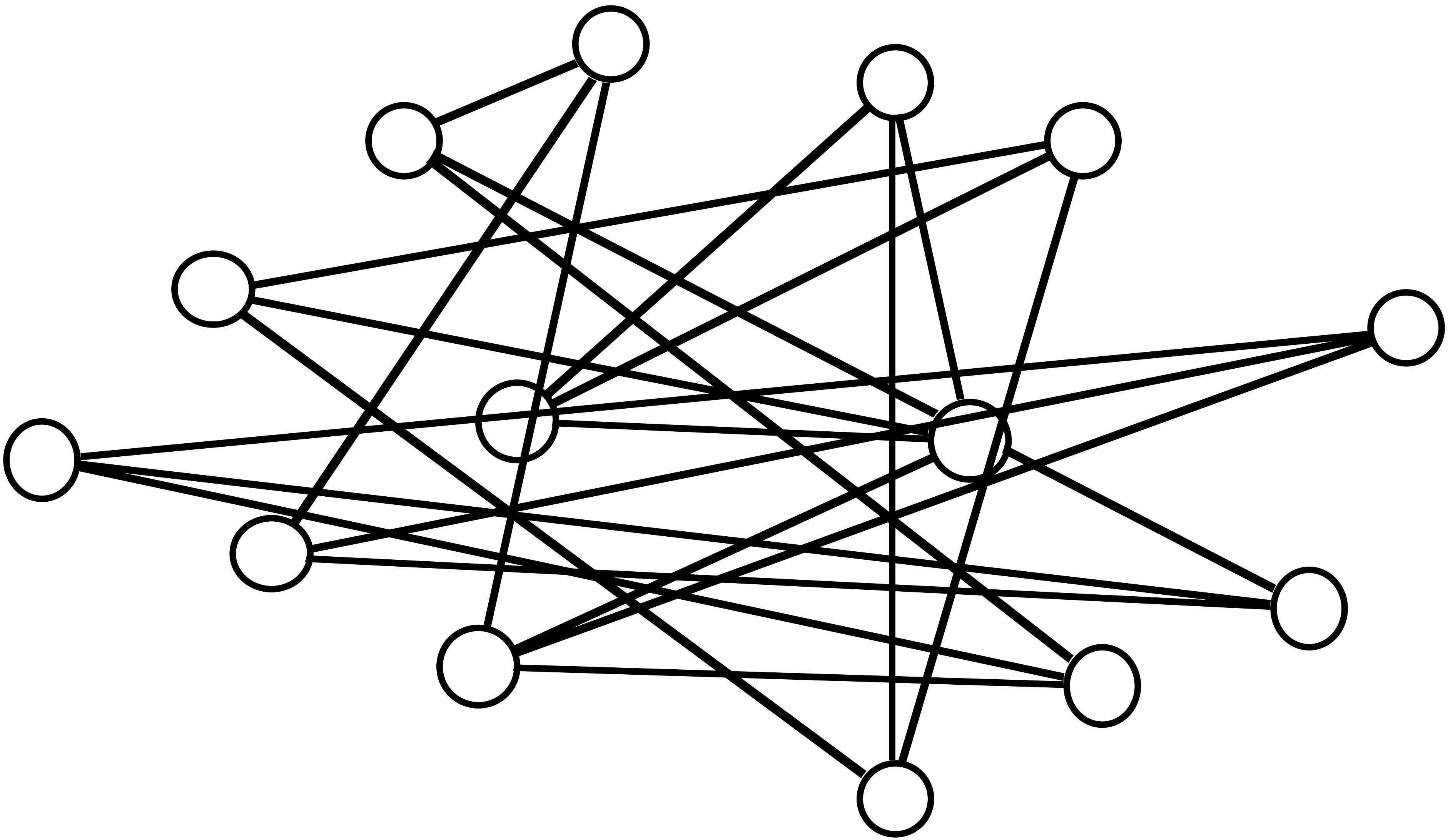
- Yes, there is a math-based science based on ideas of successive processes
- …and, Yes, there is a lot to learn:—
  - Is there an optimum number of layers?
  - If so, under what conditions?
  - Can we determine the "goodness" of an architecture?
  - Can we "measure" one architecture as "better" than another?
- Examining clustered architectures may give a clue


Clustered
Architecture
Queen Hatshepsut—XVIIIth Dynasty, 1583-1438
—Funerary Palace Architecture

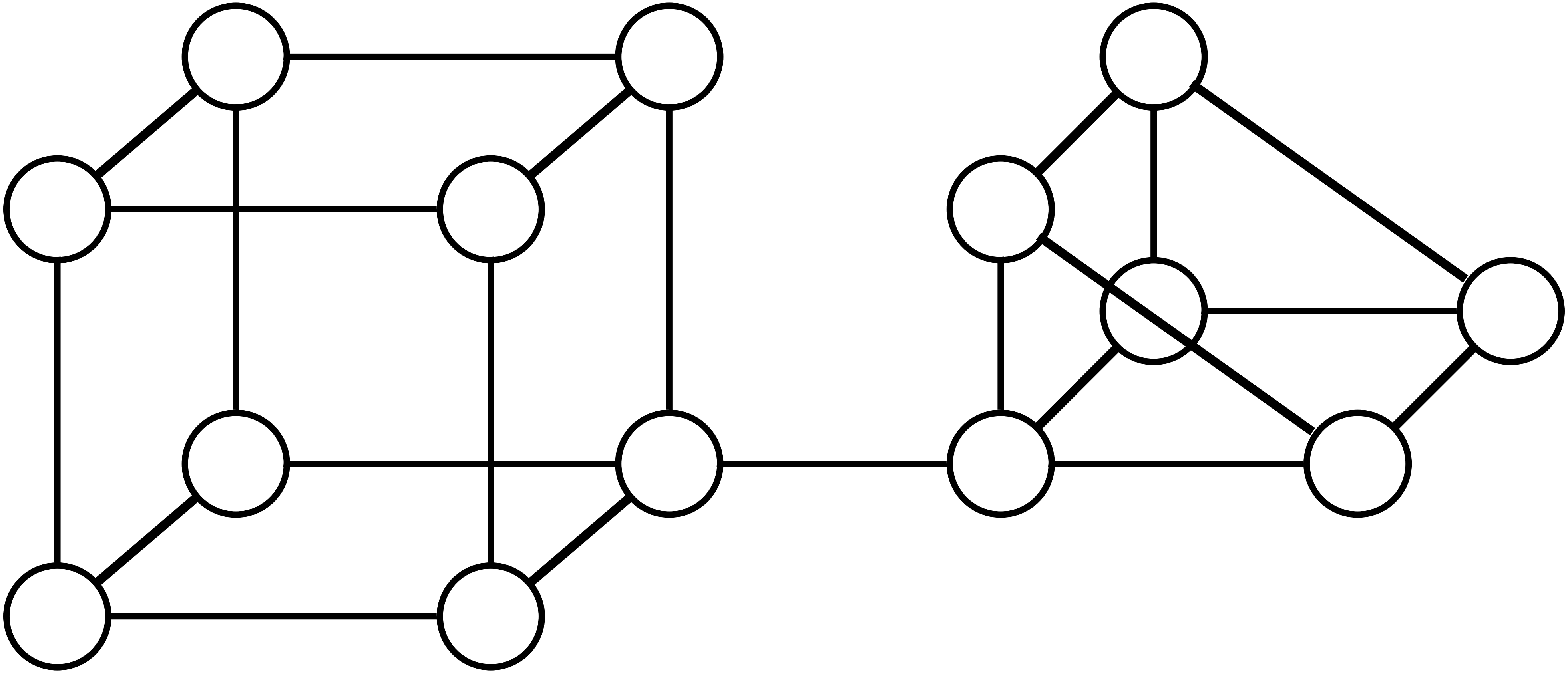
Essence of Systems Architecture

- **Moving two parts closer extends other links—**
  - there must be some optimum arrangement for *overall* performance
- **Systems Architecture** design—finding **optimum for whole system**, not just some parts

Configuration Entropy—related entities



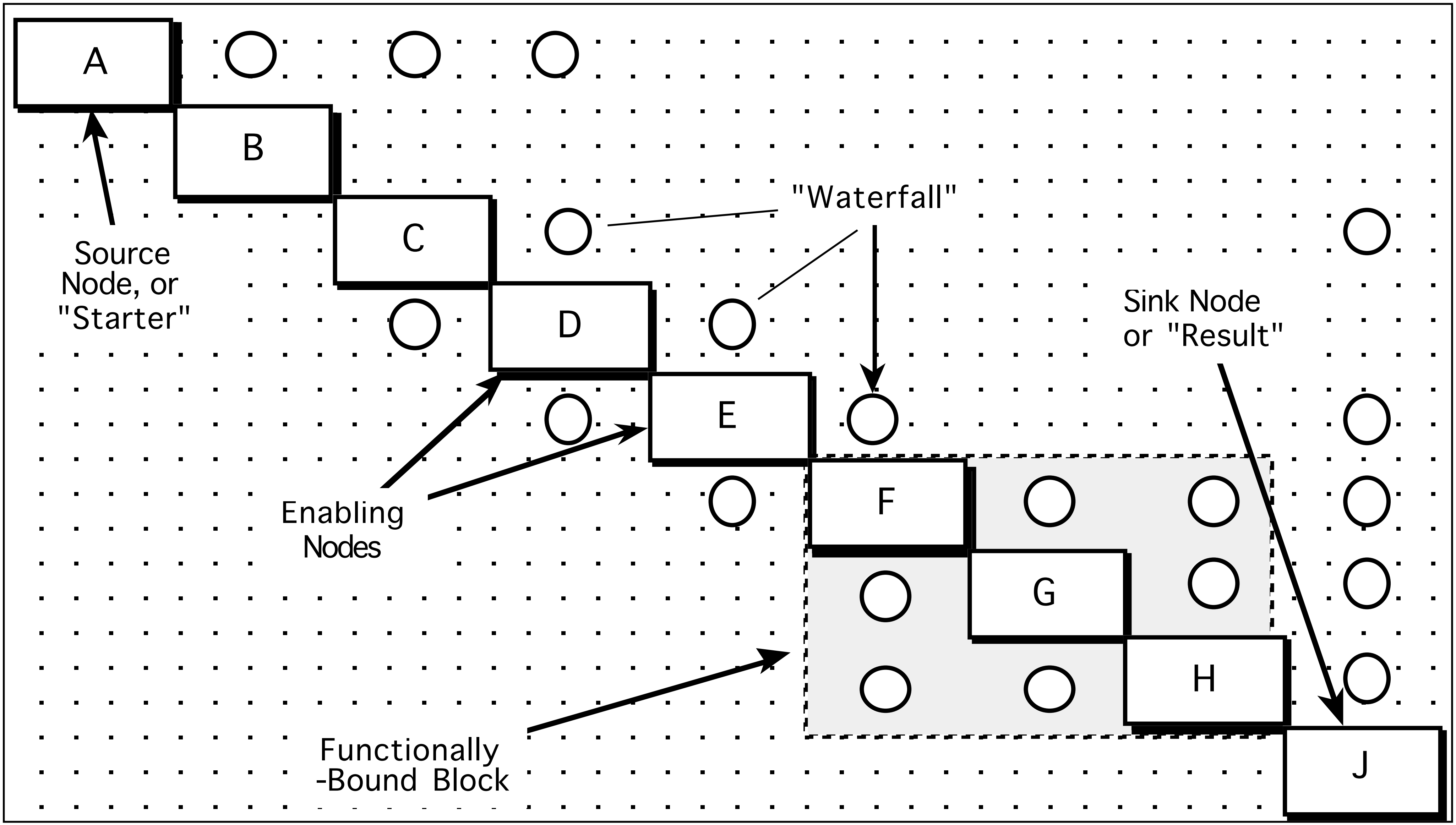
Configuration Entropy—clustered, ordered, related entities



- entity-link topology unaltered
- reduced disorder associated with reduction in overall link-length
- use this idea to *evolve architectures*

- The underlying essence of “system” is ***order***
  - ‘dent in fabric of entropy”
- So, may be able to measure the “degree of system-ness” in units of entropy—or neg-entropy?
  - lower entropy, greater “system-ness”
- Reducing system configuration entropy groups related entities into clusters, tightens the clusters
- C^2/C^3 designers familiar with this through ubiquitous N^2 Charts

The N2 Chart


A
B
C
D
E
F
G
H
J
Source Node, or "Starter"
"Waterfall"
Sink Node or "Result"
Enabling Nodes
Functionally -Bound Block

N2 and Entropy

- Internal energy trapped within high entropy system—free to escape from low entropy system
  - organizational, management, CIS implications
- Entropy determined by number of ways entities can be arranged ($2^N - 1$)
- N^2 chart can be **scored** to determine configuration entropy—the degree of disorder in the interaction pattern
- N^2 chart can be **evolved** using **genetic algorithms** to derive **minimum-entropy** pattern
- Minimum-entropy pattern ↔ optimum clustered architecture

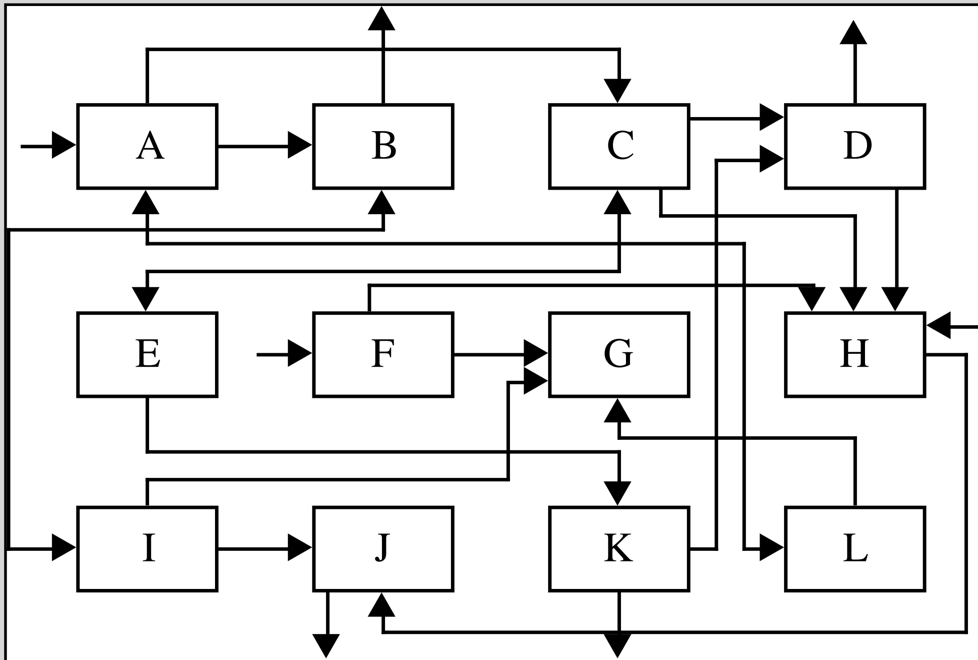
Practical Example of Clustering

Following example archetypical of many organizational and CIS/Networking Issues:—

- C^2 Ops HQ comprises 12 cells— Intel, Situation Assessment, Logistics, communications, etc., on rectangular floor.
- Individual tasks engaging C^2 Ops HQ require one, two or more cells to respond in sequence, according to type.
- Pattern of tasks uneven, some types occurring more than others
- Cell staffs co-operate/co-ordinate by walking between cells
- Rectangular room only suitable space available.

Can anything be done to reduce overall response times by rearranging cell layouts?

C² Ops HQ Example—before

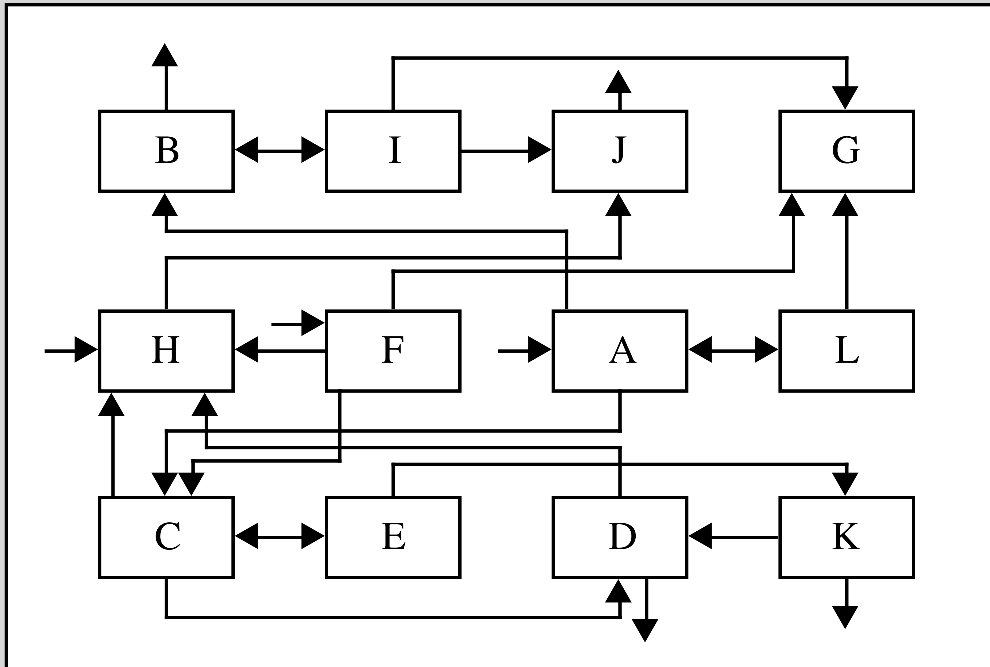


- Figure shows rectangular room with 12 cells, A—L, and arrows showing principal workflow paths

A	1	1									3
	B							2			
		C	1	1			3				
			D				2				
		3		E						3	
					F	1	1				
						G					
							H		1		
	1					2		I	2		
									J		
			3							K	
1						2					L

- Matrix represents path-lengths between cells A—L. Numbers represent path utilization e.g. 1 = low, 2 = moderate, 3 = heavy
- Work index $=\Sigma_i$ (Path-length $_i$ * Utilization $_i$) for i = 1 to 12
- **<u>Work index from matrix = 160</u>**

C² Ops HQ Example—after



- Figure shows cells rearranged to maintain original work-flow logic, but reduce overall work Index
- Paths form "waterfall"

```
B 2
1 I 2 2
    J
      G
      2 L 1
1       3 A     1
      1     F 1
    1         H
              3 C 1 1
                3 E   3
              2     D
                    3 K
```

- Matrix score = f(Entropy)
- Some separations increased, e.g. A to B, but overall path-length reduced from 79 to 36, i.e. by 54%
- Matrix rearranged to reduce overall value of Work Index by 65% in the work of communicating between cells
- <u>**New Work Index = 56**</u>

Synthesizing Architecture

- C^2 Ops HQ example shows practical advantages of clustering.
- Genetic Clustering approach:—
  - accumulates and analyses data
  - maintains over view of whole, as aggregation, not just of parts (machines), but of interactions between all parts (material exchanges)
  - enables optimization of whole, rather than of each part piecemeal
  - hard numbers—real, measurable results
  - breadth of application limited only by imagination of user

…Offers basis for
***auto-adaptive* CIS/C4i architectures……**

Conclusion

- Architecture not generally recognized as design parameter
- Increasing system complexity emphasizes value of optimal architecture
- Goal of sound architecture:—
  - simpler, more efficient, more effective system
  - adaptable, damage-tolerant, sustainable performance
- Systems architecture amenable to rigorous scientific study

CIS community should adopt architecture as central, formal design subject for hardware, software, systems, organization, processing, networking, auto-adaptation…