

# Fiefdoms to Kingdoms

And Parallels from Stovepipes to an Enterprise

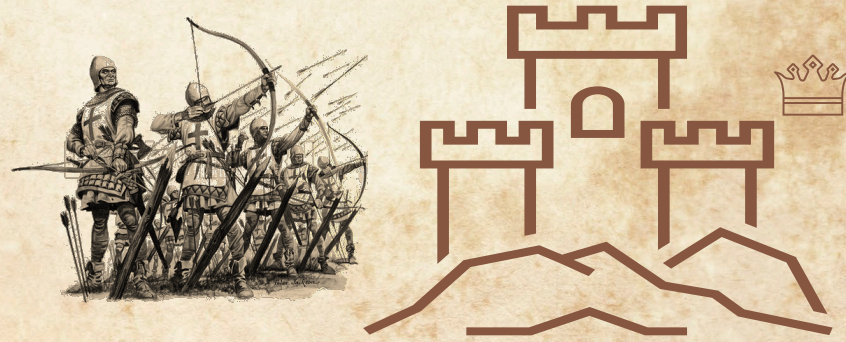


Gerry Simon – Chief Architect / Mission Solutions / Parsons

© 2023 by Parsons, Published by The Aerospace Corporation with permission.

# Fiefdoms to a Kingdom

Service	Kingdom Service Catalog			
Weaponry				
Culture				
Governance				
Infrastructure				

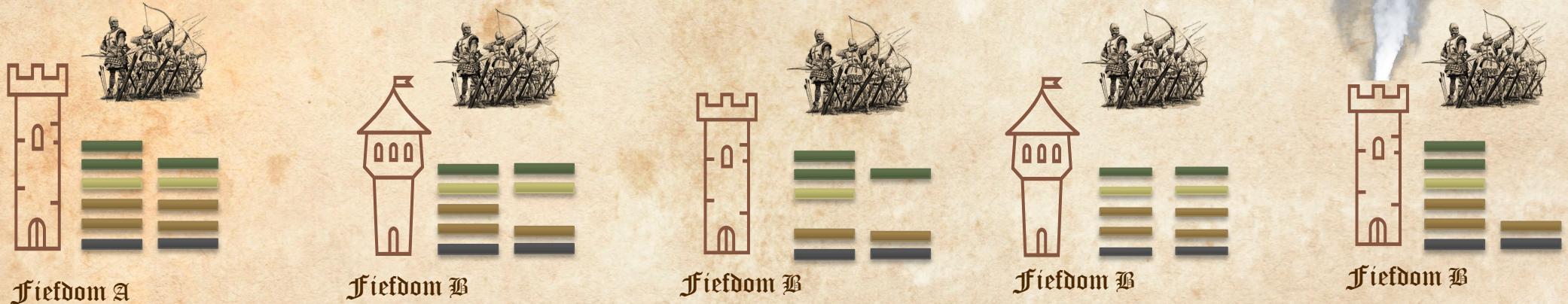


## Fiefdoms

Incompatible weapons  
Mismatched tactics  
No shared infrastructure  
Lose Wars

## Kingdoms

Shared Weaponry Development  
Shared Tactics  
Roads and Ports  
Win Wars



Fiefdom A

Fiefdom B

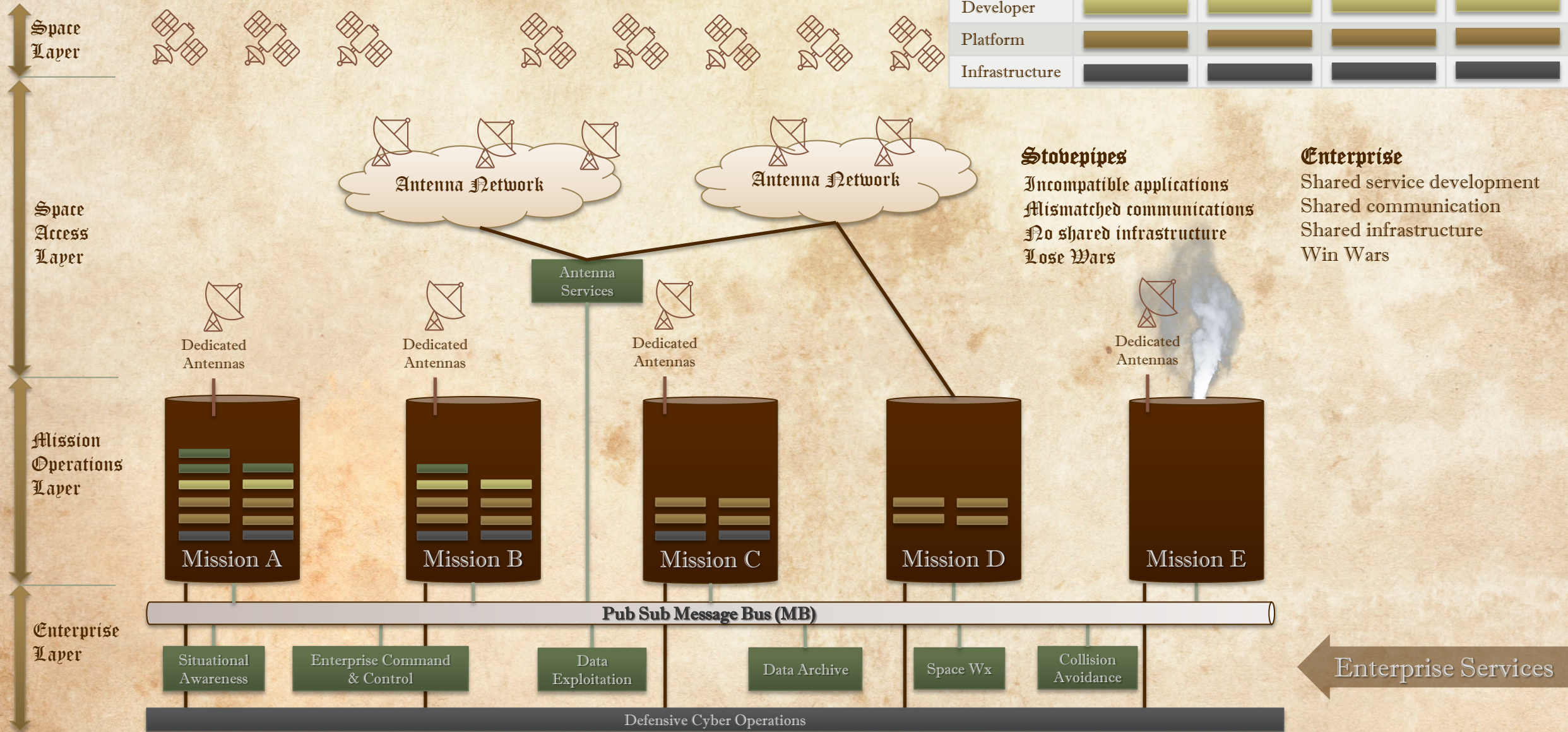
Fiefdom B

Fiefdom B

Fiefdom B



# Stovepipes to a Ground Enterprise



# Backup

$$\frac{dI}{dt} \geq \frac{d^2N}{dt^2}$$

# Questions

- Degrees of interoperability at a ground enterprise level can be accomplished through various means, including:
  - A common IT infrastructure (centralized or distributed)
    - Sure. Reduces cost, forces some interoperability, but not essential
  - Common software services or code base
    - NO! Too far, may even be bad, CMSF
  - A common user experience (human machine interface)
    - OK, but don't limit innovation
  - Common data and messaging standards
    - YES, This is **the most** important
  - Common CONOPS across programs
    - OK, but don't limit innovation
  - Standardized spacecraft systems, components and/or interfaces
- Top Two challenges
  - The Lords don't want to be in a Kingdom
  - Messaging: Creation, governance, sustainment

# Innovation Patterns Enable Accelerated Innovation

- **Disruptive Technology**

- New technology that creates new solutions to old problems
- **Examples:** Cloud computing, Automated Test and Deployment, Software Defined Radio, Commercial Antenna Networks

- **Key standards**

- Key technical standards provide a foundation innovation can grow from
- **Examples:** C2MS, CCSDS Link Standards, XTCE, GCMS

- **Investment**

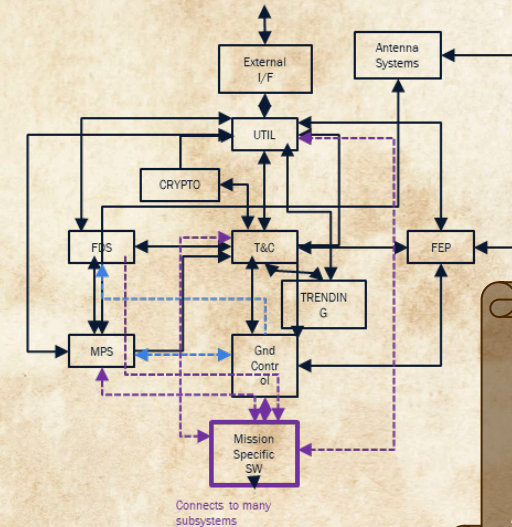
- Market must exist for new capability

- **Disruptive Thinking**

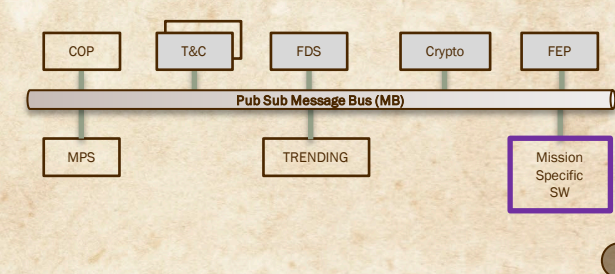
- New approaches to old problems or simply quit working old problems
- **Examples:** DevSecOps, Ruthlessly drive out touch labor, and things I haven't thought of

- **Competition**

- Competition rewards cost effective and innovative capability



Traditional Tightly Coupled Architecture



Modern Service Based Architecture