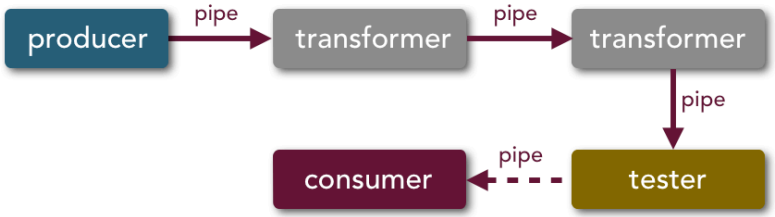
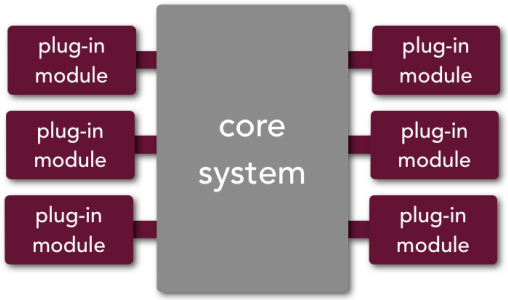
Architecture Patterns 2

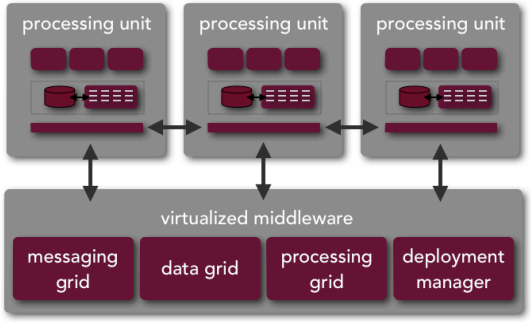
architecture patterns 2



pipeline architecture



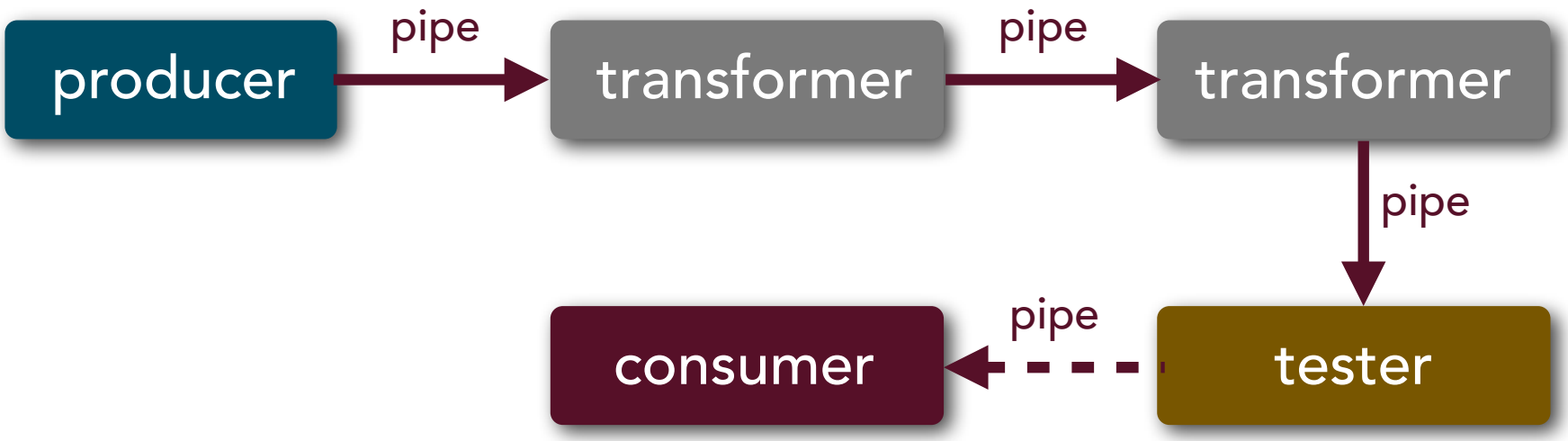
microkernel architecture



space-based architecture

pipeline architecture

(a.k.a. pipe and filter architecture)



pipeline architecture

architectural components

pipe

filter

pipeline architecture

pipes

uni-directional only

usually point-to-point for high performance, but could be message-based for scalability

payload can be any type (text, bytes, object, etc.)

pipeline architecture

filters

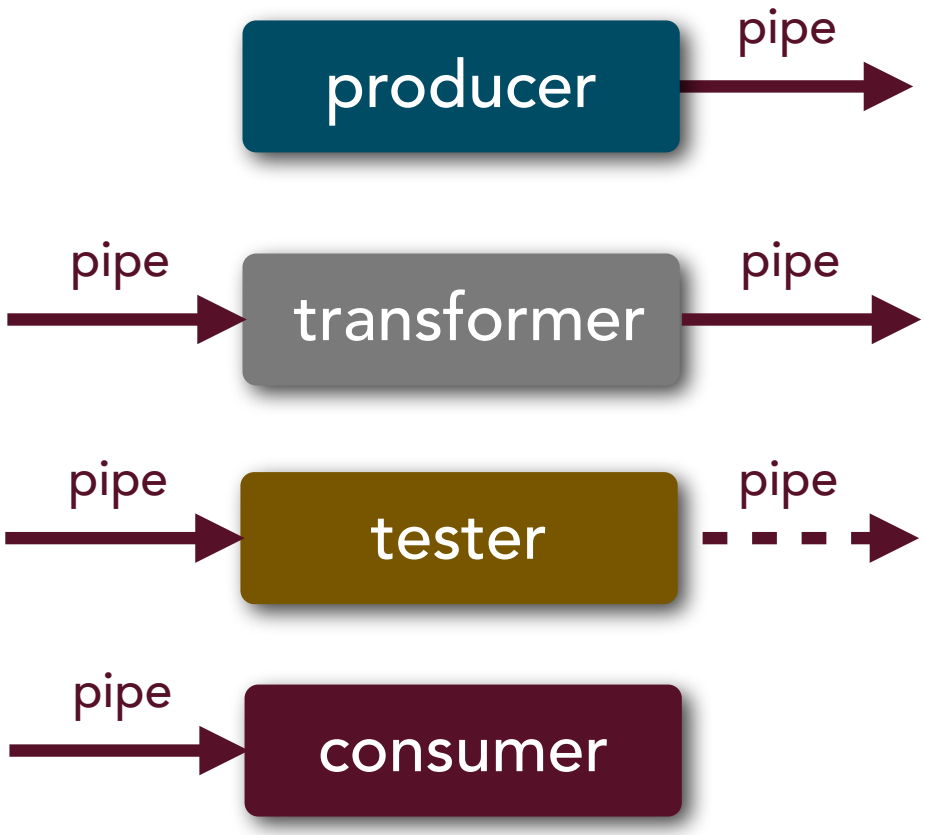
self-contained and independent from other filters

usually designed to perform a single specific task

four filter types (producer, consumer, transformer, and tester

pipeline architecture

filters



starting point, outbound only

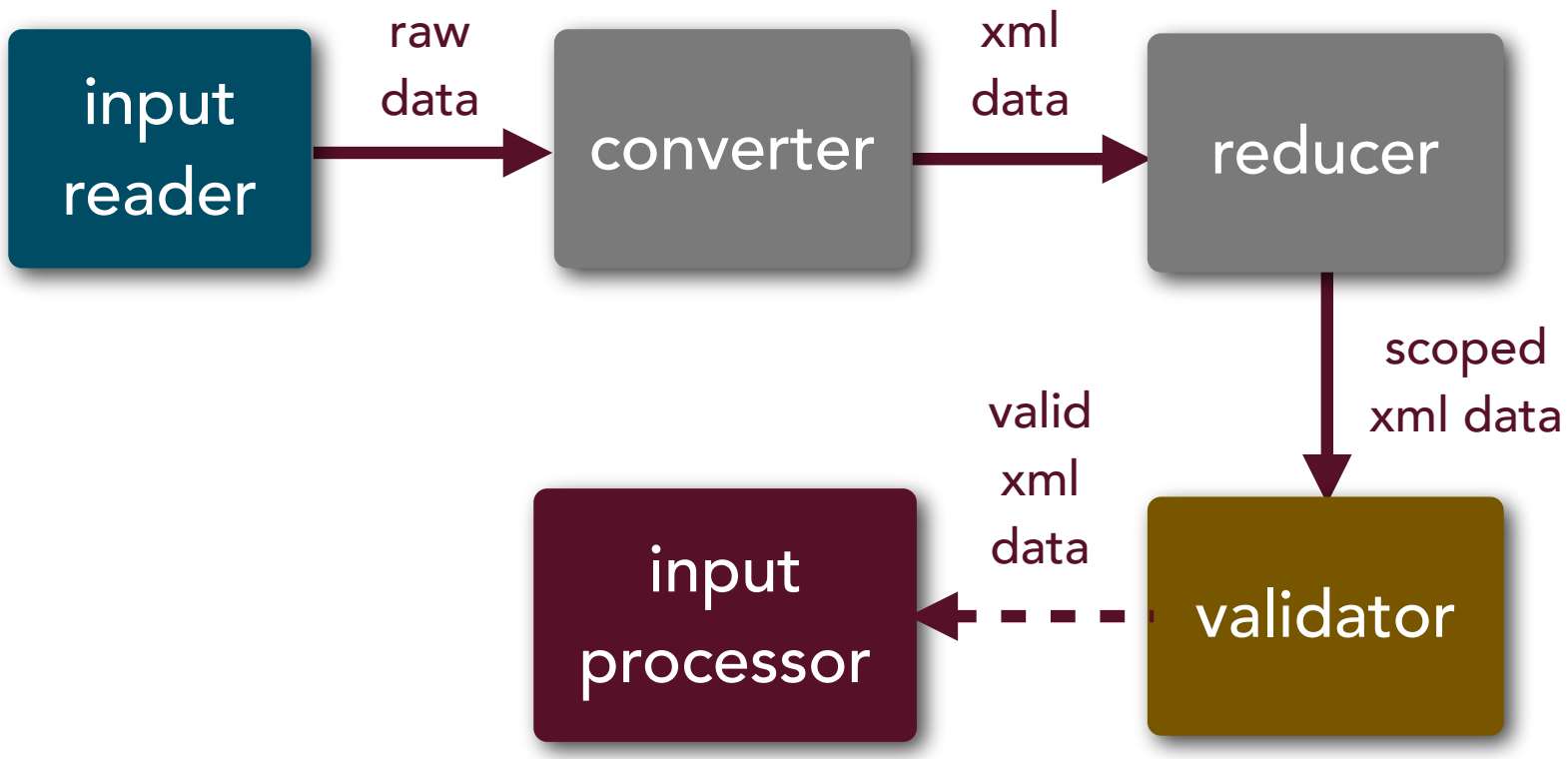
input, processing, output

input, discard or pass-thru

ending point, inbound only

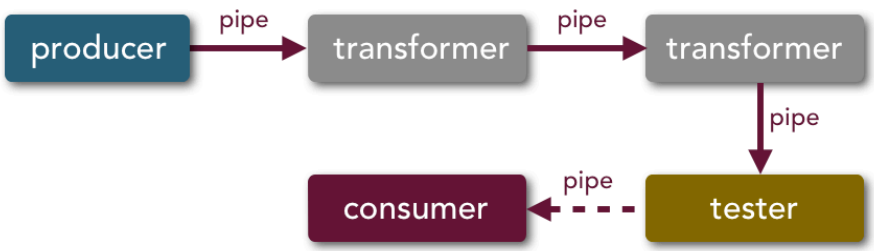
pipeline architecture

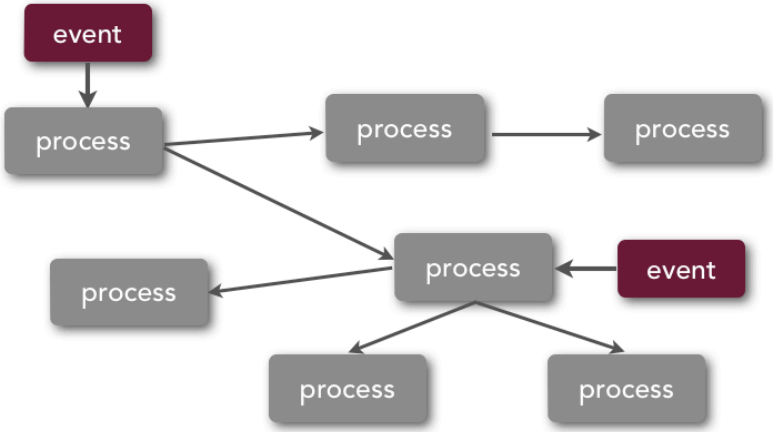
example



pipeline architecture

pipeline vs. event-driven

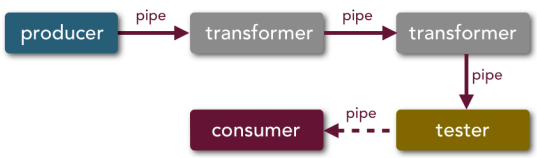


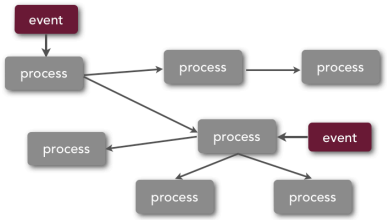


aren't these really the same pattern?

pipeline architecture

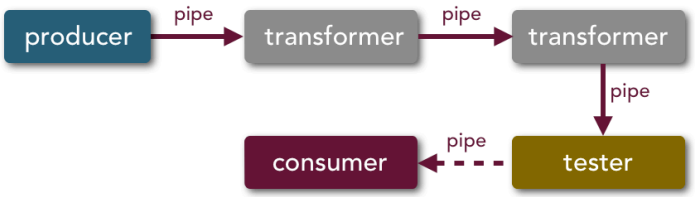
pipeline vs. event-driven





synchronous data filtering	asynchronous event processing
single target for pipe	multiple targets for event
simple single purpose filters	complex multi-purpose processors

pipeline architecture



useful for smaller deterministic systems with a distinct processing flow

filters can easily be added and removed

provides for a high level of decoupling

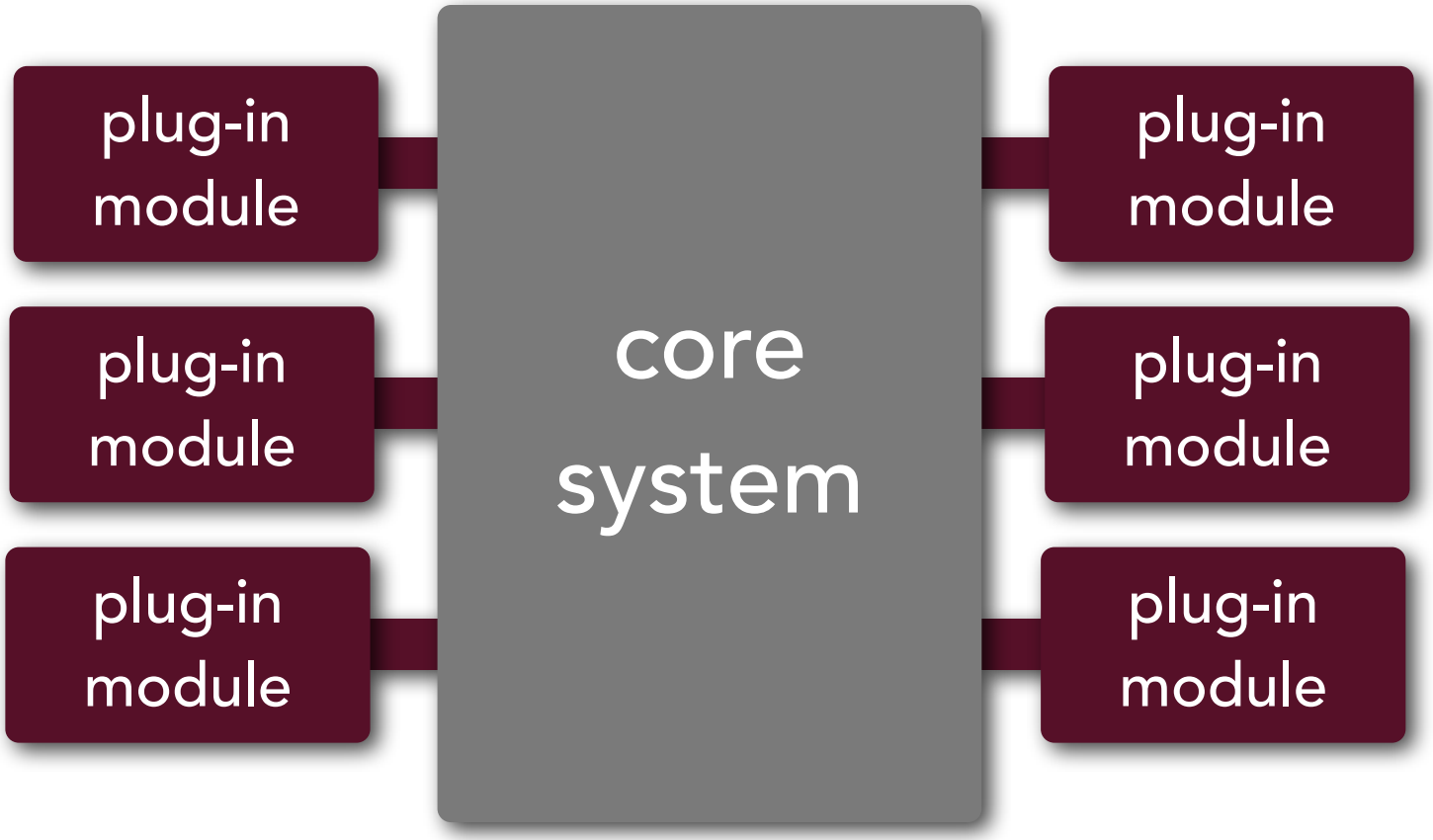
supports evolutionary design

able to easily adapt to changing requirements

can easily be incorporated into another pattern

microkernel architecture

(a.k.a. plug-in architecture pattern)



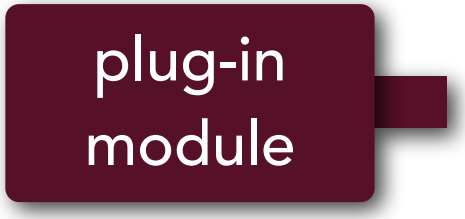
microkernel architecture

architectural components

minimal functionality to run system

general business rules and logic

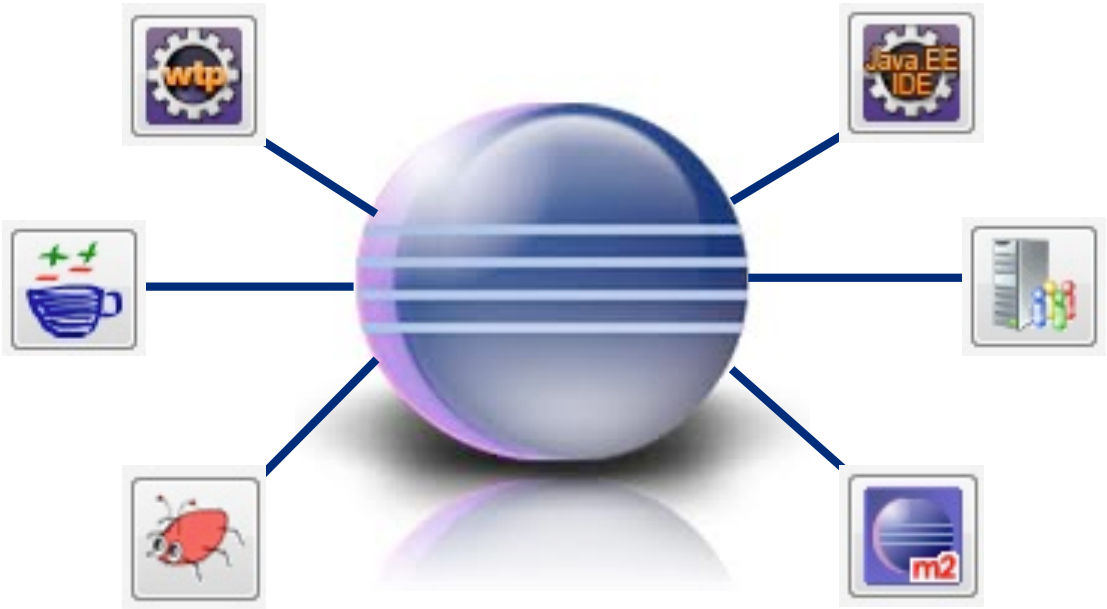
doesn't contain custom processing



standalone independent module

specific additional rules or logic

microkernel architecture

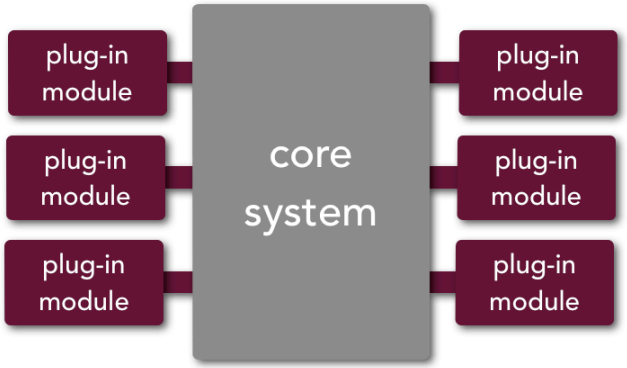


microkernel architecture

claims processing

microkernel architecture

useful for systems that have custom processing or processing is susceptible to change



plug-in modules can easily be added and removed

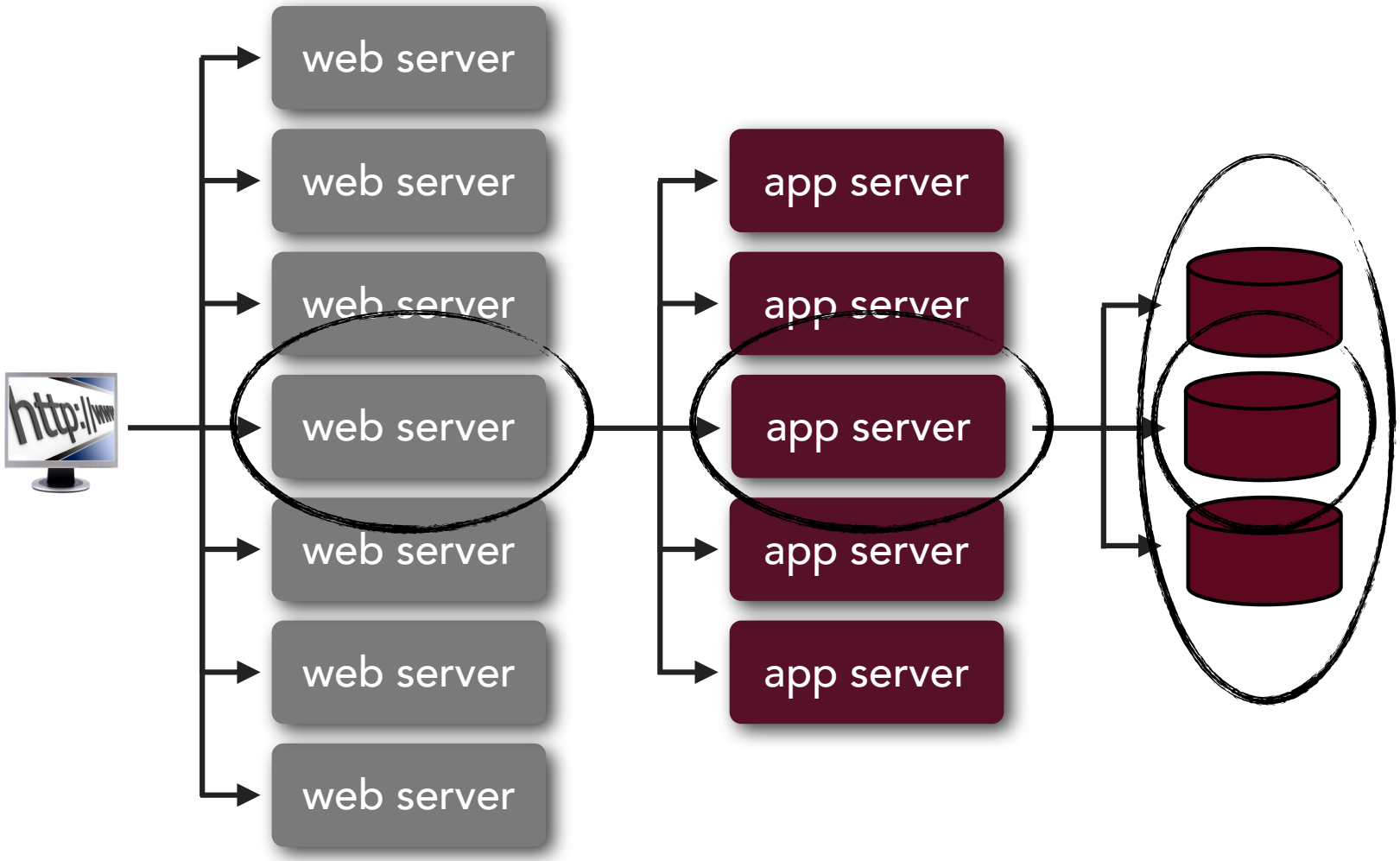
supports evolutionary design

able to easily adapt to changing requirements

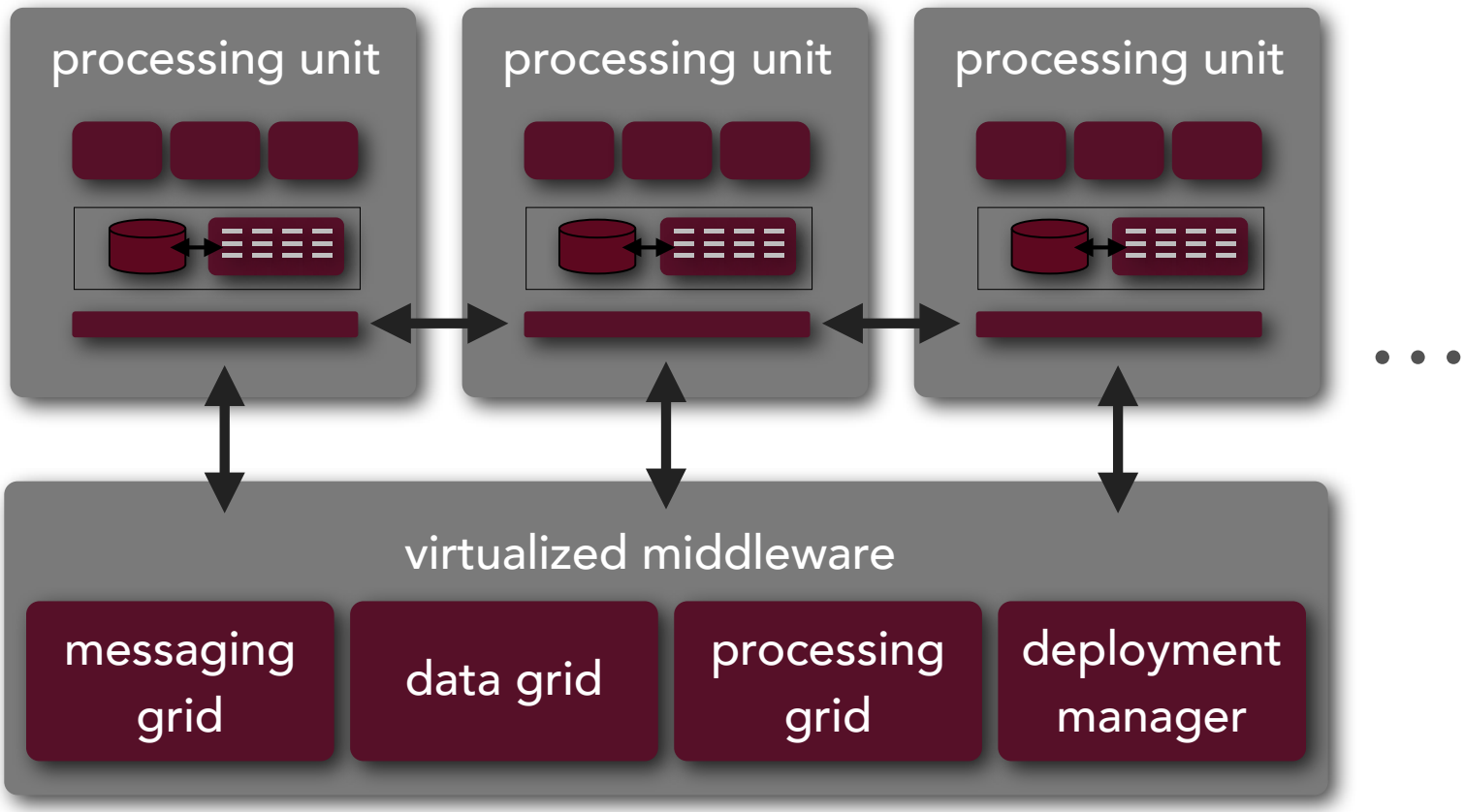
can easily be incorporated into another pattern

space-based architecture

let's talk about scalability for a moment...

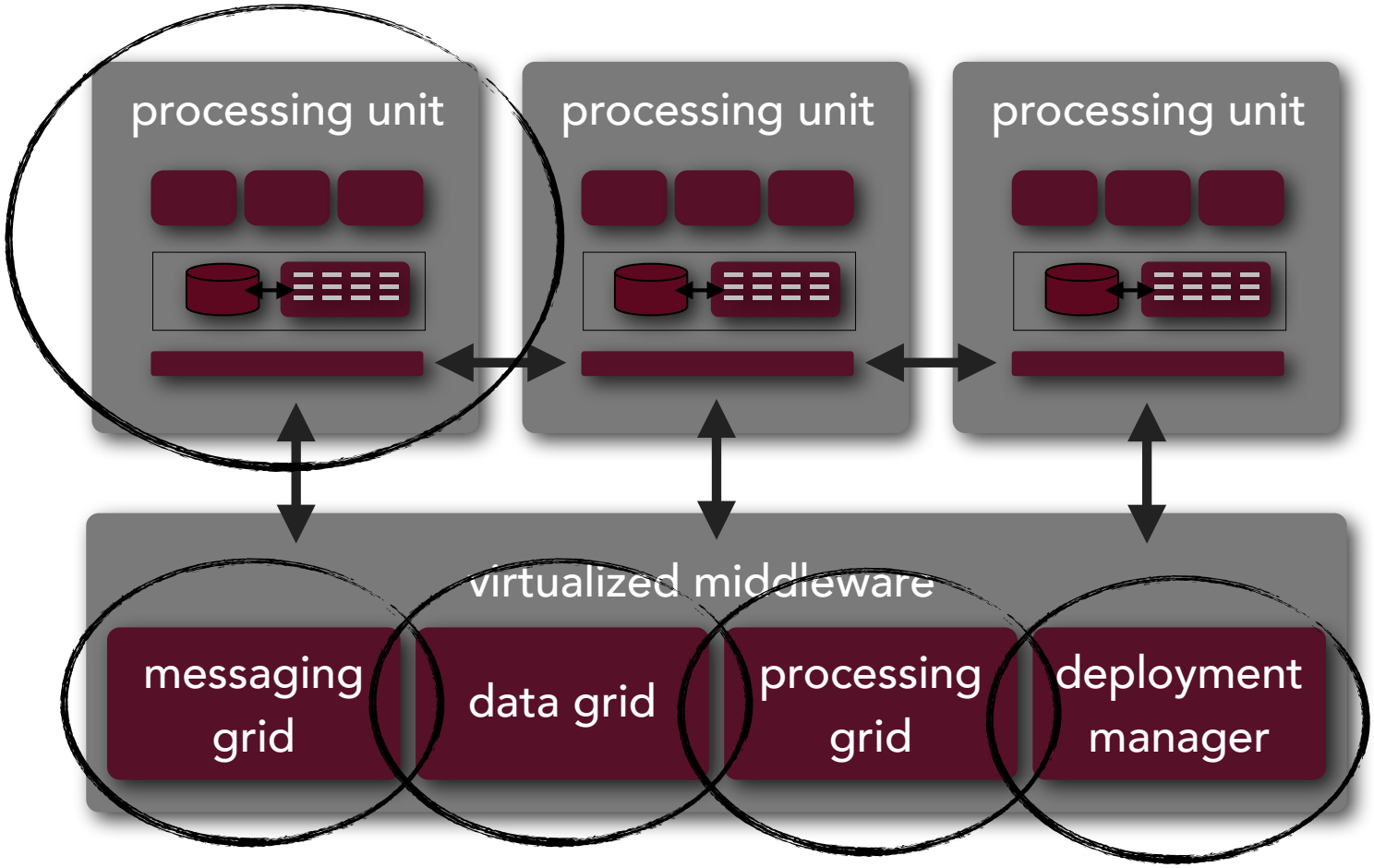


space-based architecture



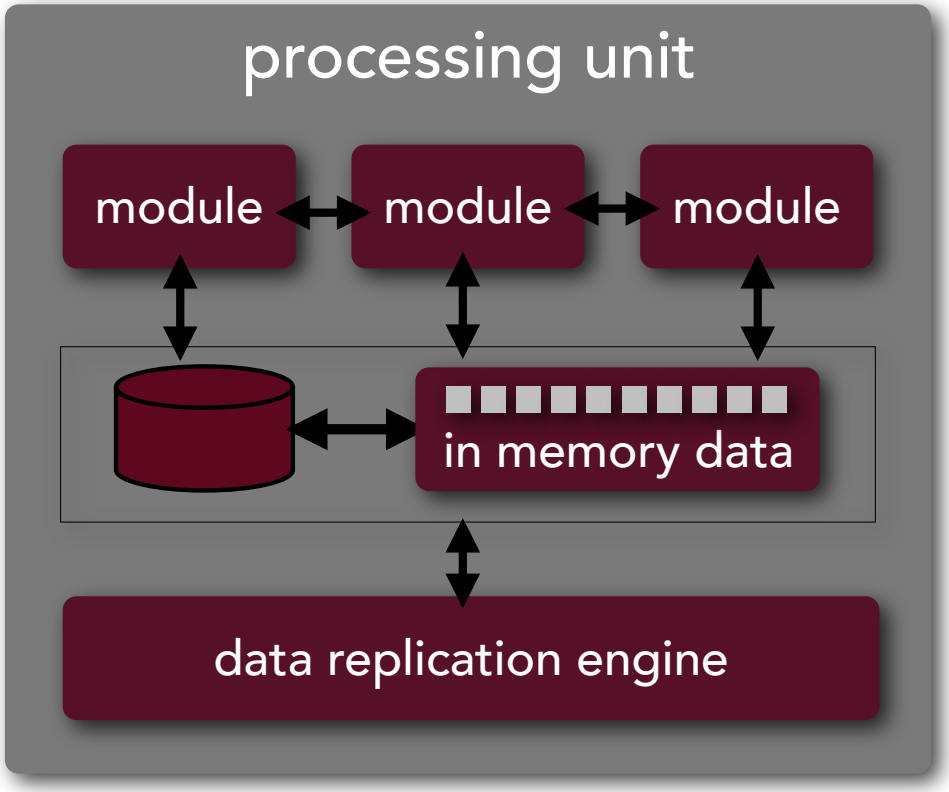
space-based architecture

architectural components



space-based architecture

processing unit



space-based architecture

middleware

messaging grid

data grid

processing grid

deployment manager

space-based architecture

middleware

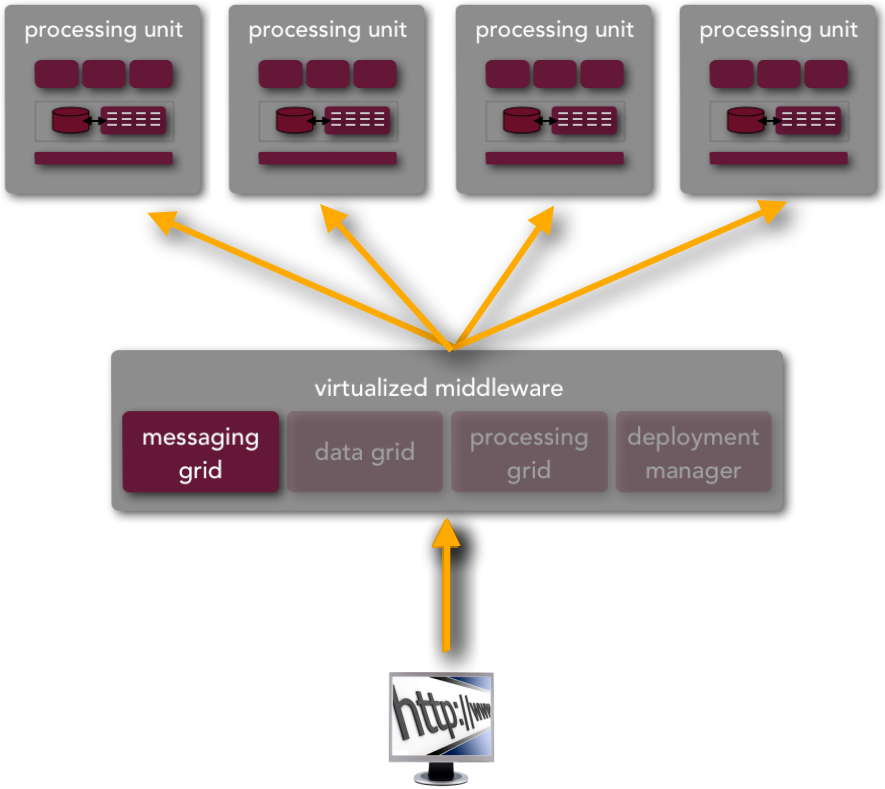
messaging grid

data grid

processing grid

deployment manager

manages input request and session



space-based architecture

middleware

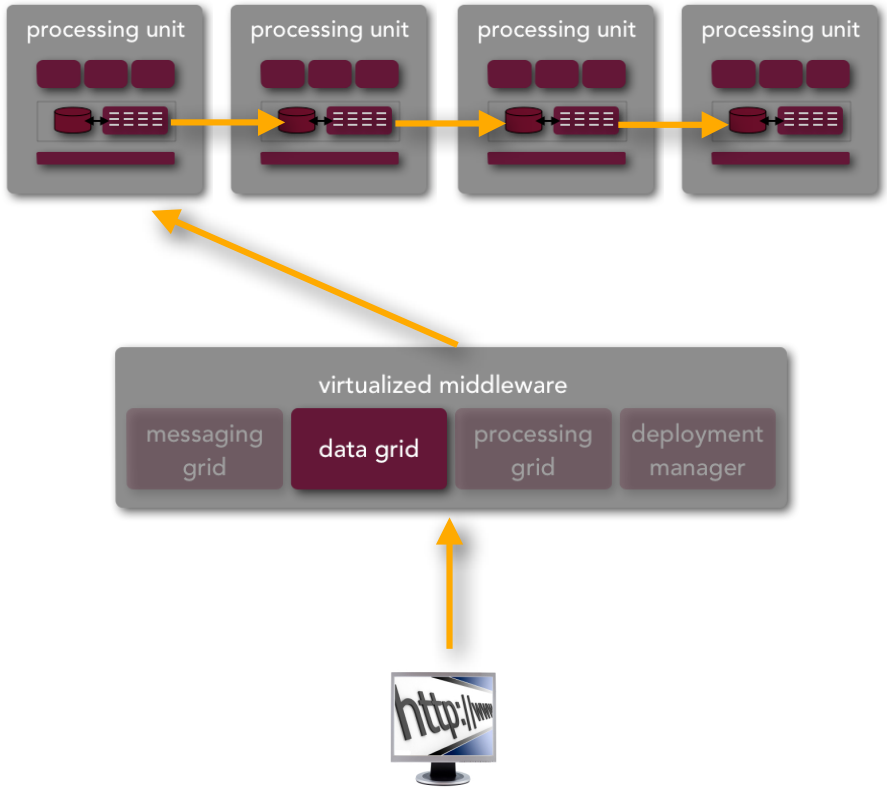
manages data replication between processing units

messaging grid

data grid

processing grid

deployment manager



space-based architecture

middleware

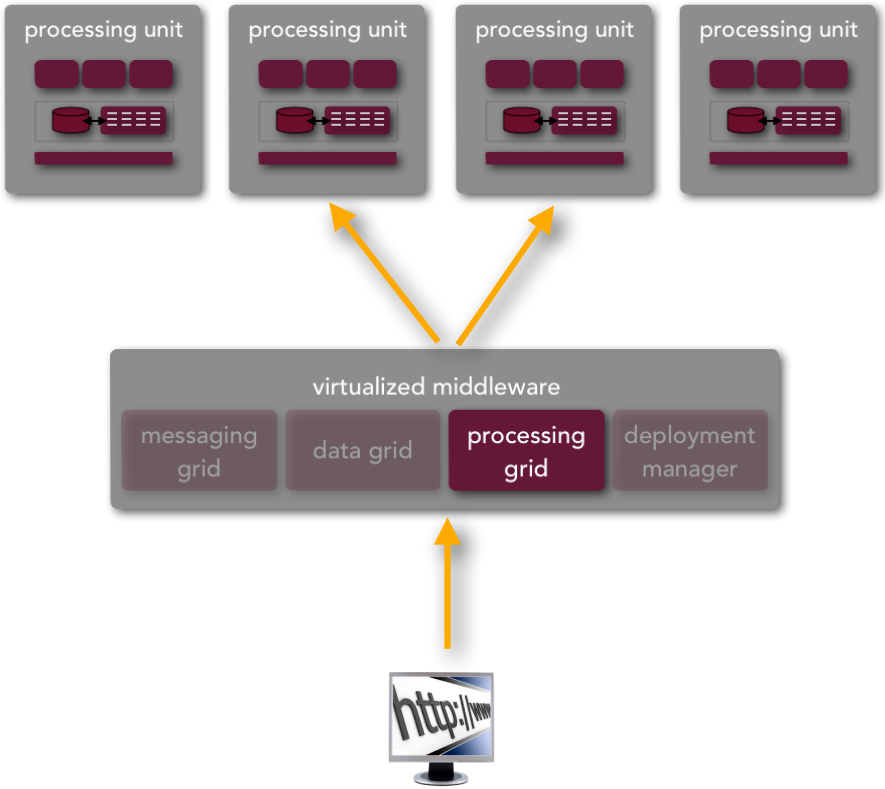
messaging grid

data grid

processing grid

deployment manager

manages parallel request processing



space-based architecture

middleware

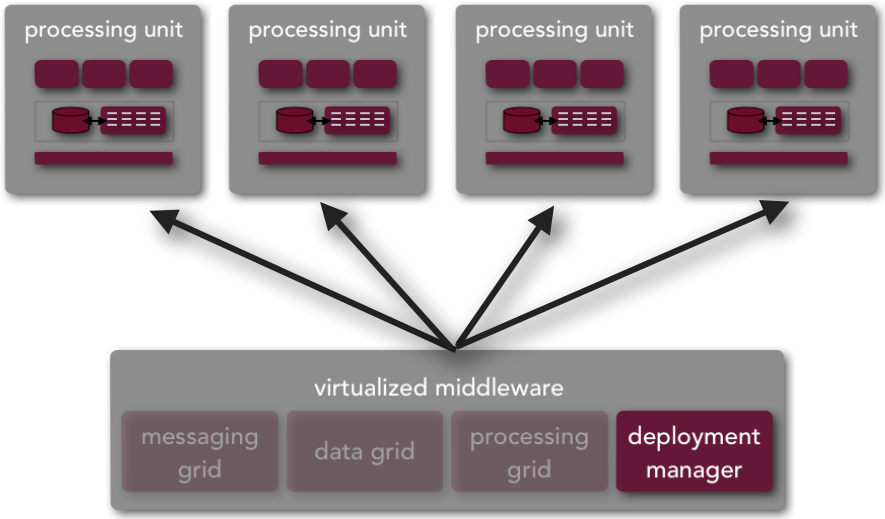
manages dynamic processing unit deployment

messaging grid

data grid

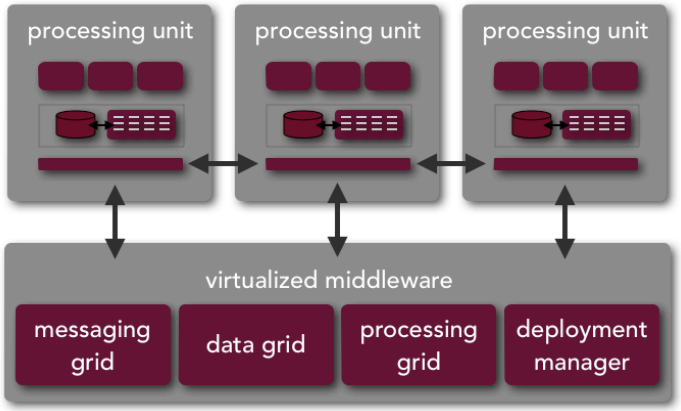
processing grid

deployment manager



space-based architecture

it's all about variable scalability...



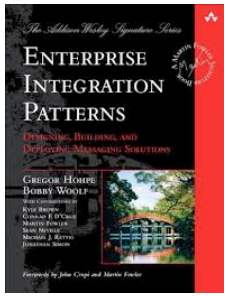
good for applications that have variable load or inconsistent peak times

not a good fit for traditional large-scale relational database systems

relatively complex and expensive pattern to implement

for more information

Wikipedia (space-based architecture)
http://en.wikipedia.org/wiki/Space-based_architecture



Enterprise Integration Patterns
http://www.eaipatterns.com/PipesAndFilters.html

Wikipedia (pipeline architecture)
http://en.wikipedia.org/wiki/Pipeline_(software)

?'s

Mark Richards
Independent Consultant
Hands-on Enterprise / Integration Architect
Published Author / Conference Speaker

http://www.wmrichards.com
http://www.linkedin.com/pub/mark-richards/0/121/5b9

Published Books:
Java Message Service, 2nd Edition
97 Things Every Software Architect Should Know
Java Transaction Design Strategies

Neal Ford
Director / Software Architect /
Meme Wrangler

ThoughtWorks®

2002 Summit Blvd, Level 3, Atlanta, GA 30319, USA
T: +1 40 4242 9929 Twitter: @neal4d
E: nford@thoughtworks.com W: thoughtworks.com