



Architecture AntiPatterns

Part 1

architecture anti-patterns 1



architecture by implication



gold plating



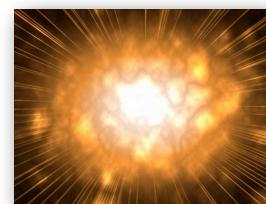
covering your assets



vendor king



witches brew



big bang architecture

architecture by implication

systems lacking a clear documented architecture



architecture by implication

what **architecture pattern** are you using?

what **client model** is most appropriate?

which **platform** is best for this solution?

does the **hardware or os** matter?

how will you handle **component integration**?

which **communication protocols** should you use?

architecture by implication

is the solution **feasible** given skills, budget, and time?

how **secure** does the system need to be?

does the system need to **scale**?

how much **performance** is needed from the system ?

how **available** does the system need to be?

do you need to be concerned about **Maintainability**?

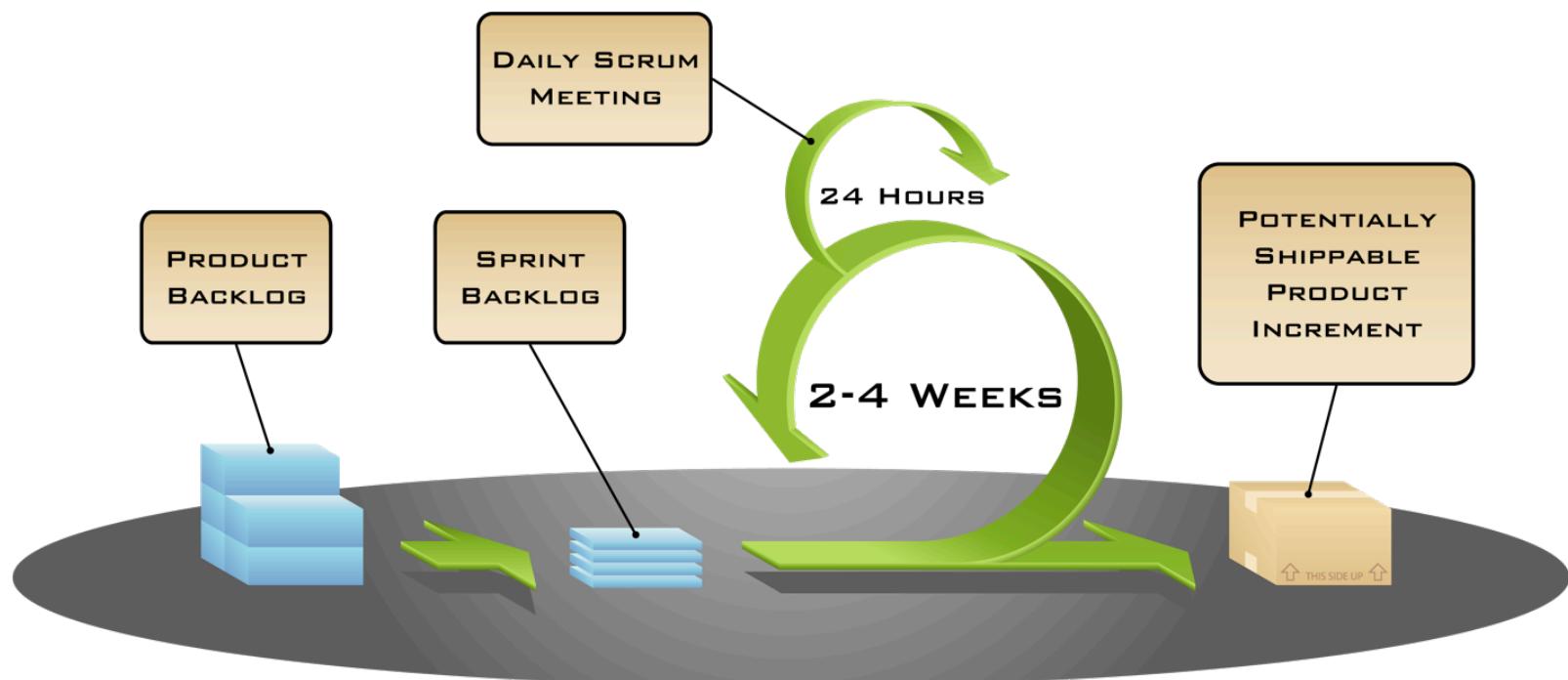
architecture by implication

be careful about over-confidence



architecture by implication

remember that agile methodologies are not a substitute for creating an architecture



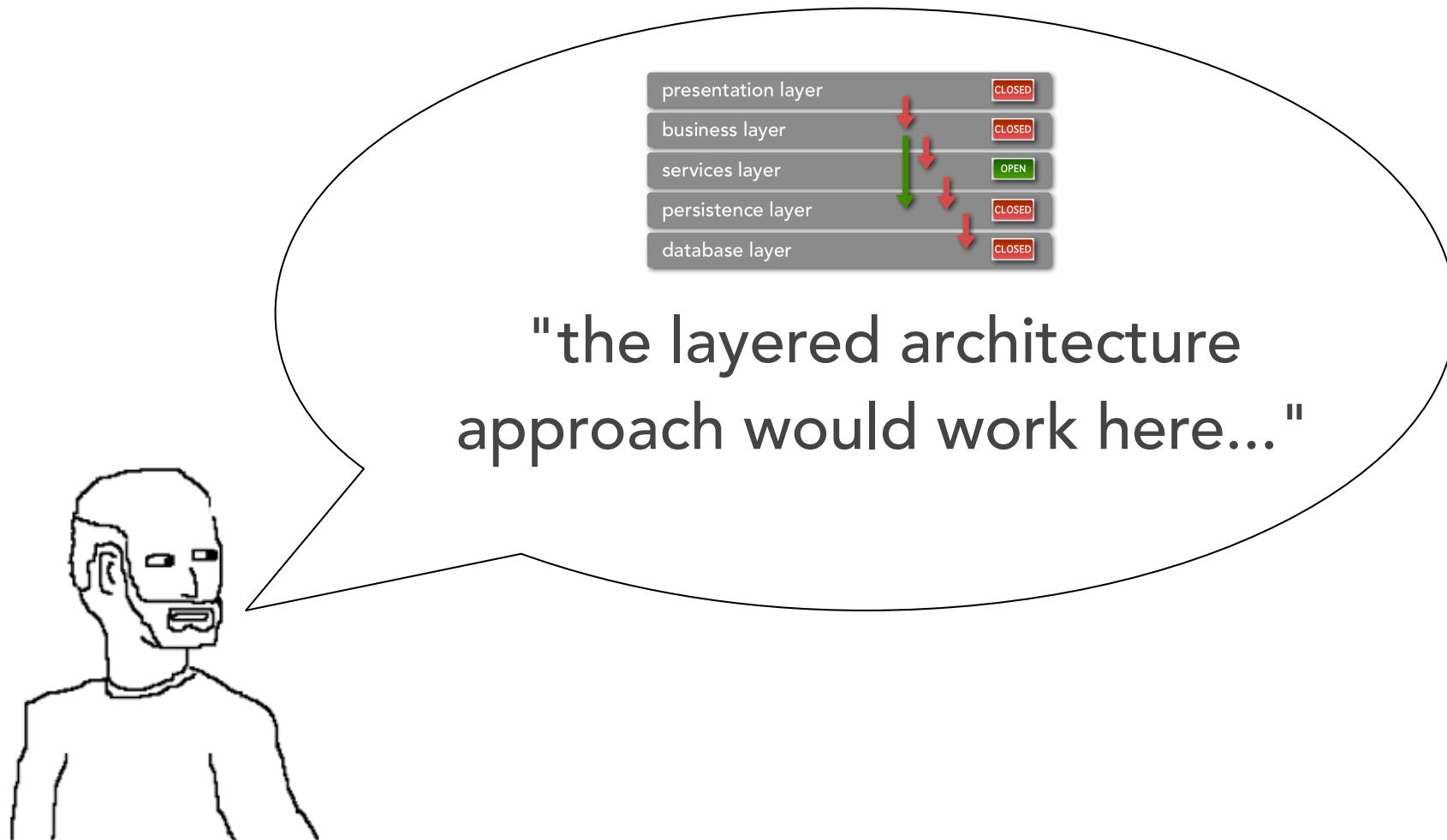
COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

cover your assets

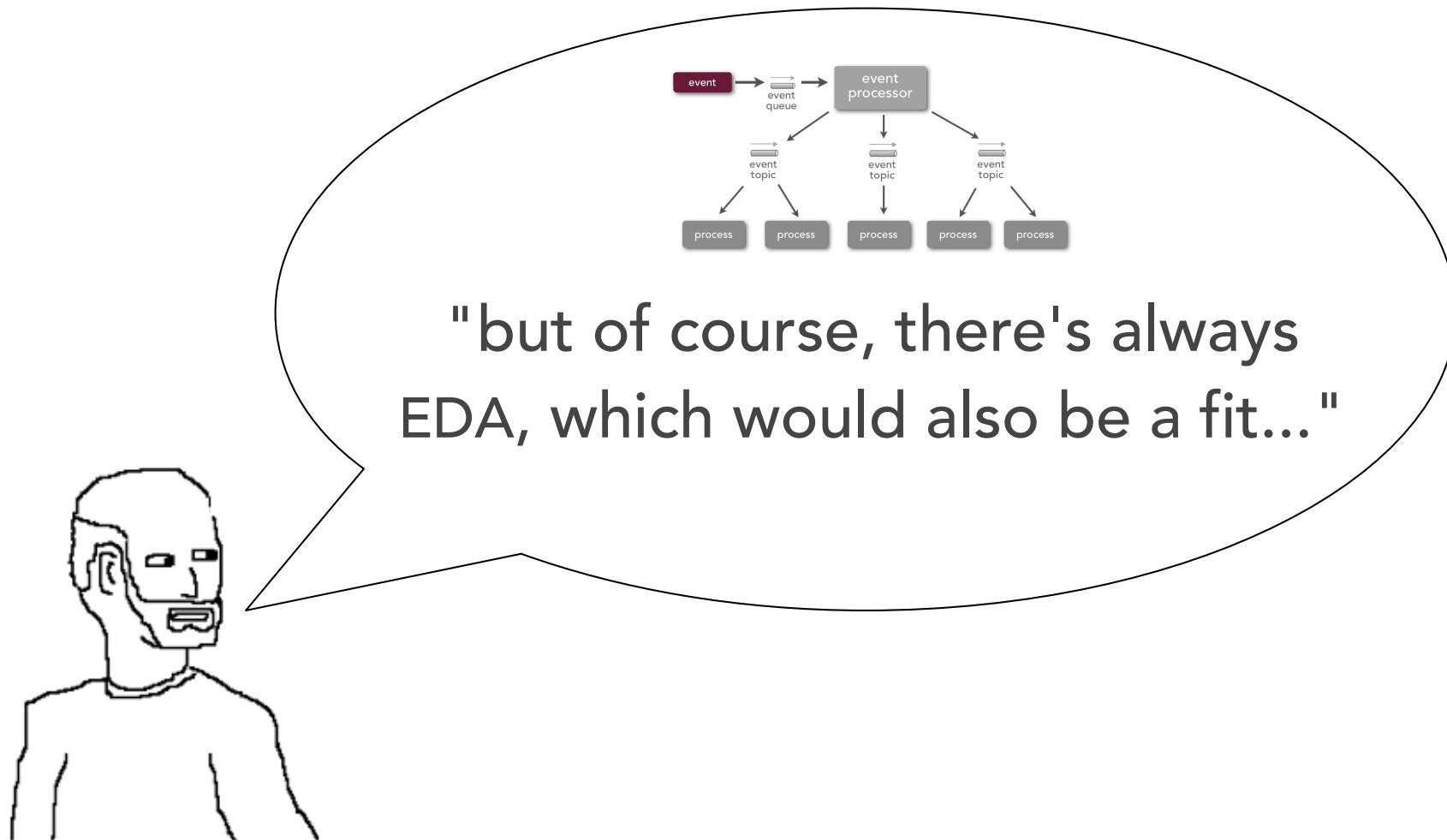
continuing to document and present alternatives
without ever making an architecture decision



cover your assets



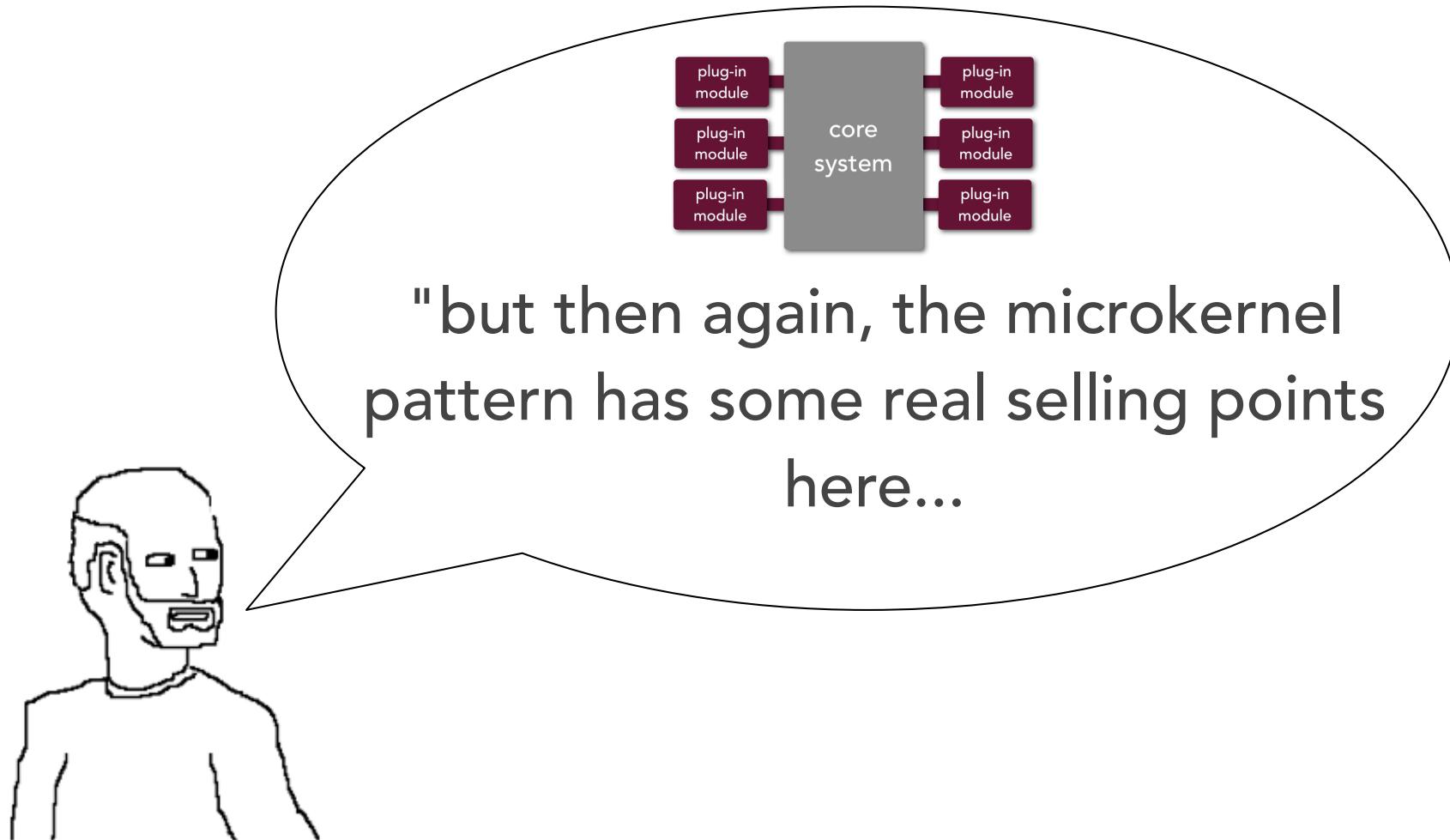
cover your assets



cover your assets

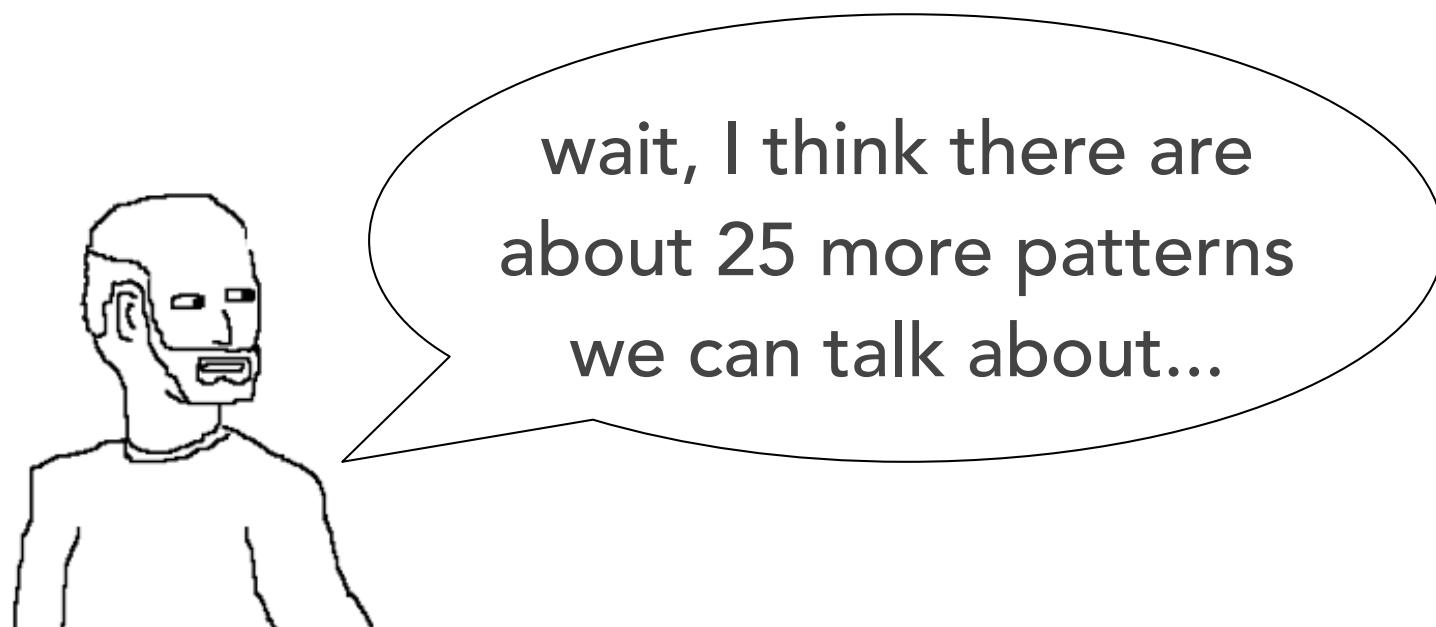


cover your assets



cover your assets

it's your job as an architect to present alternatives, clearly articulate the pros and cons of each, and recommend the best solution for the situation



witches brew

architectures are designed by groups resulting in a complex mixture of ideas and lack of clear vision



witches brew

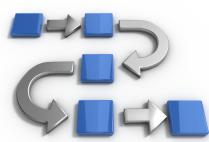




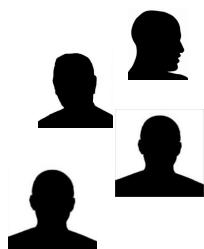
witches brew



responsible and
knowledgeable architect



architecture validation and
tradeoff analysis (A-ATAM)



ongoing architecture
discussions and peer reviews



gold plating

continuing to define the architecture well past the point which the extra effort is adding any value



gold plating



too many details hide the core principles and standards
commonly leads to the analysis paralysis anti-pattern in
that the initial architecture is never actually released

adding gold plating to a well-defined architecture
significantly increases costs with little or no value

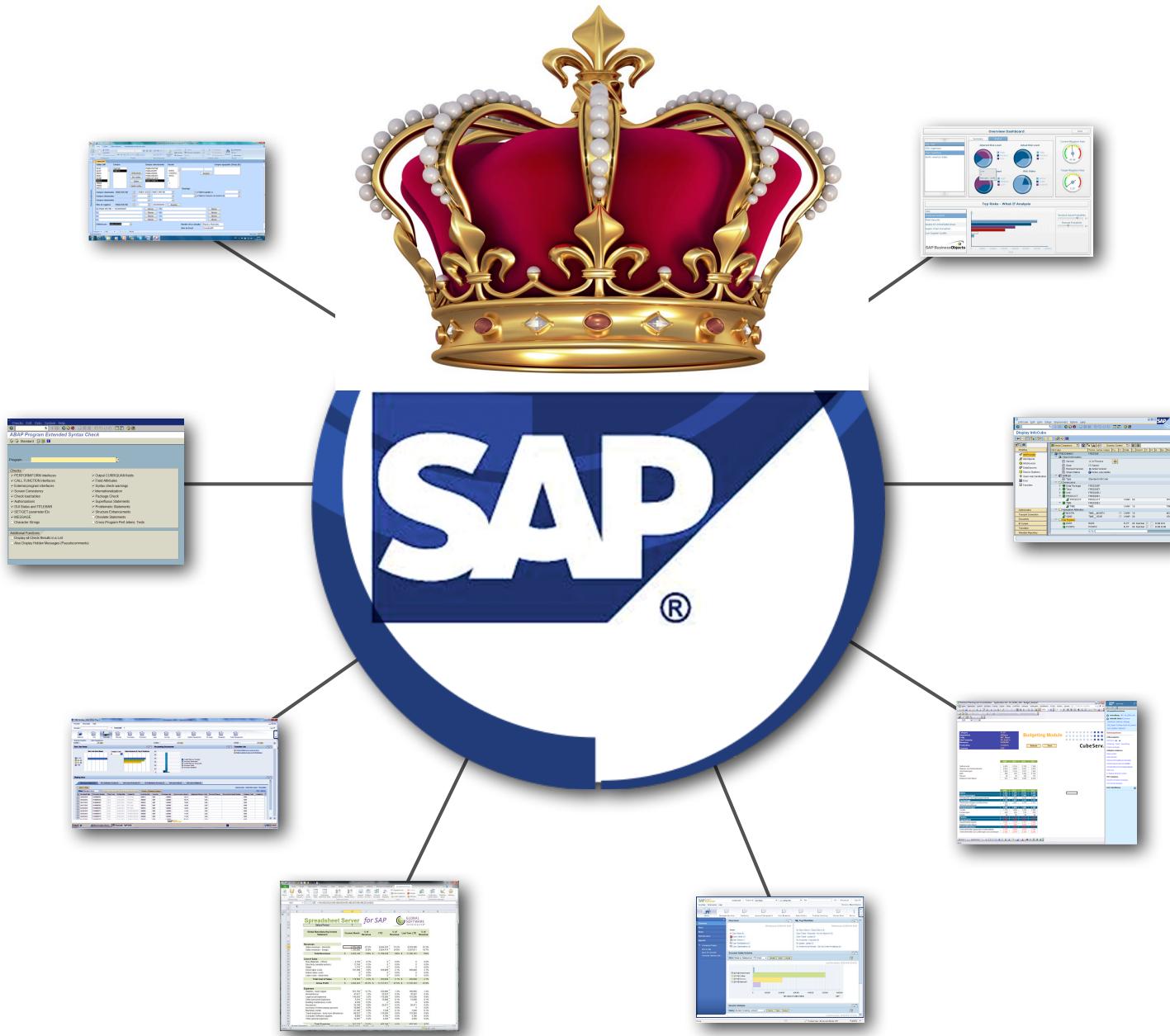
a complex and overly-detailed architecture is hard to
understand and comprehend; no "big picture"

vendor king

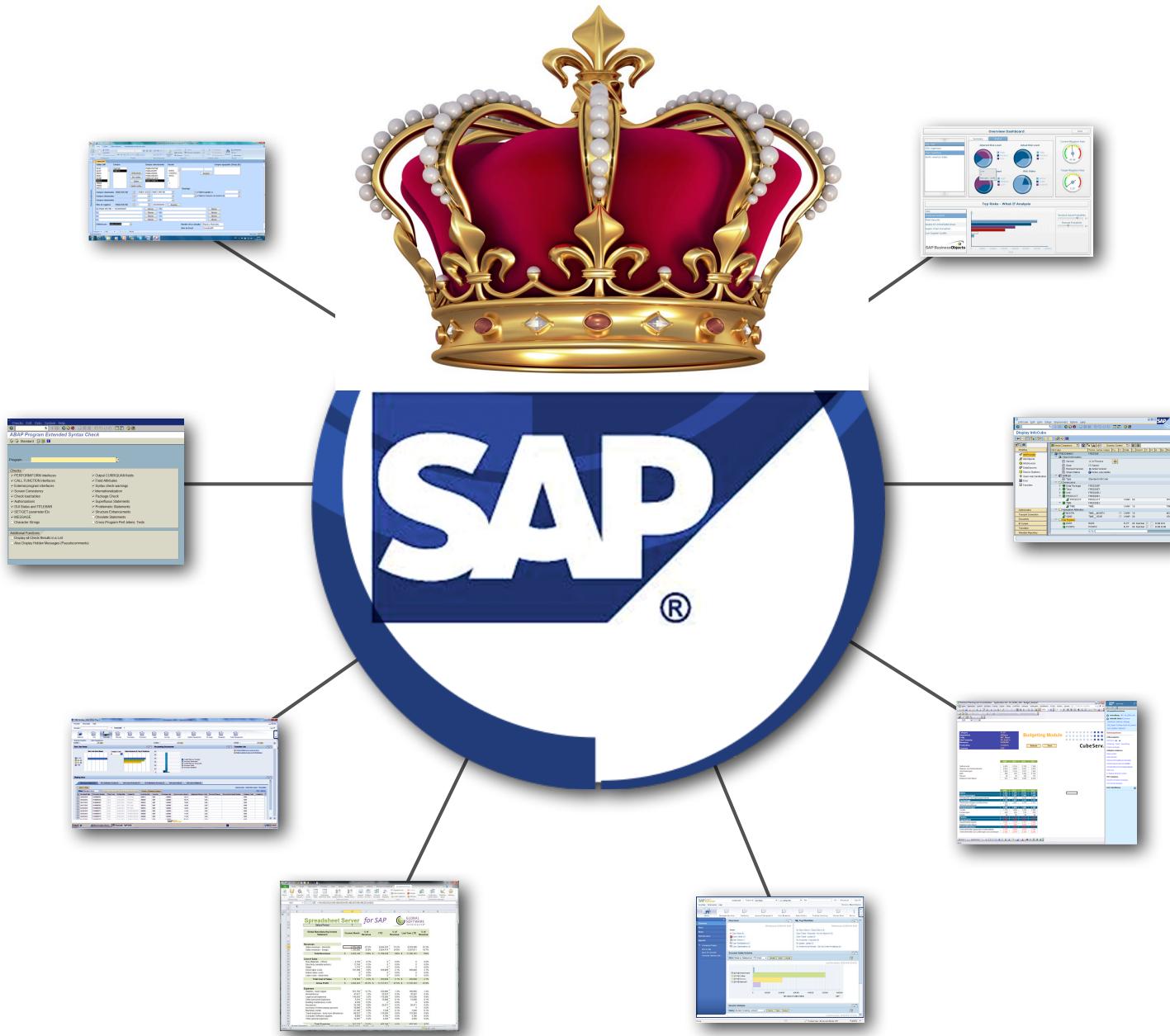
product-dependent architectures leading to a loss
of control of architecture and development costs



vendor king

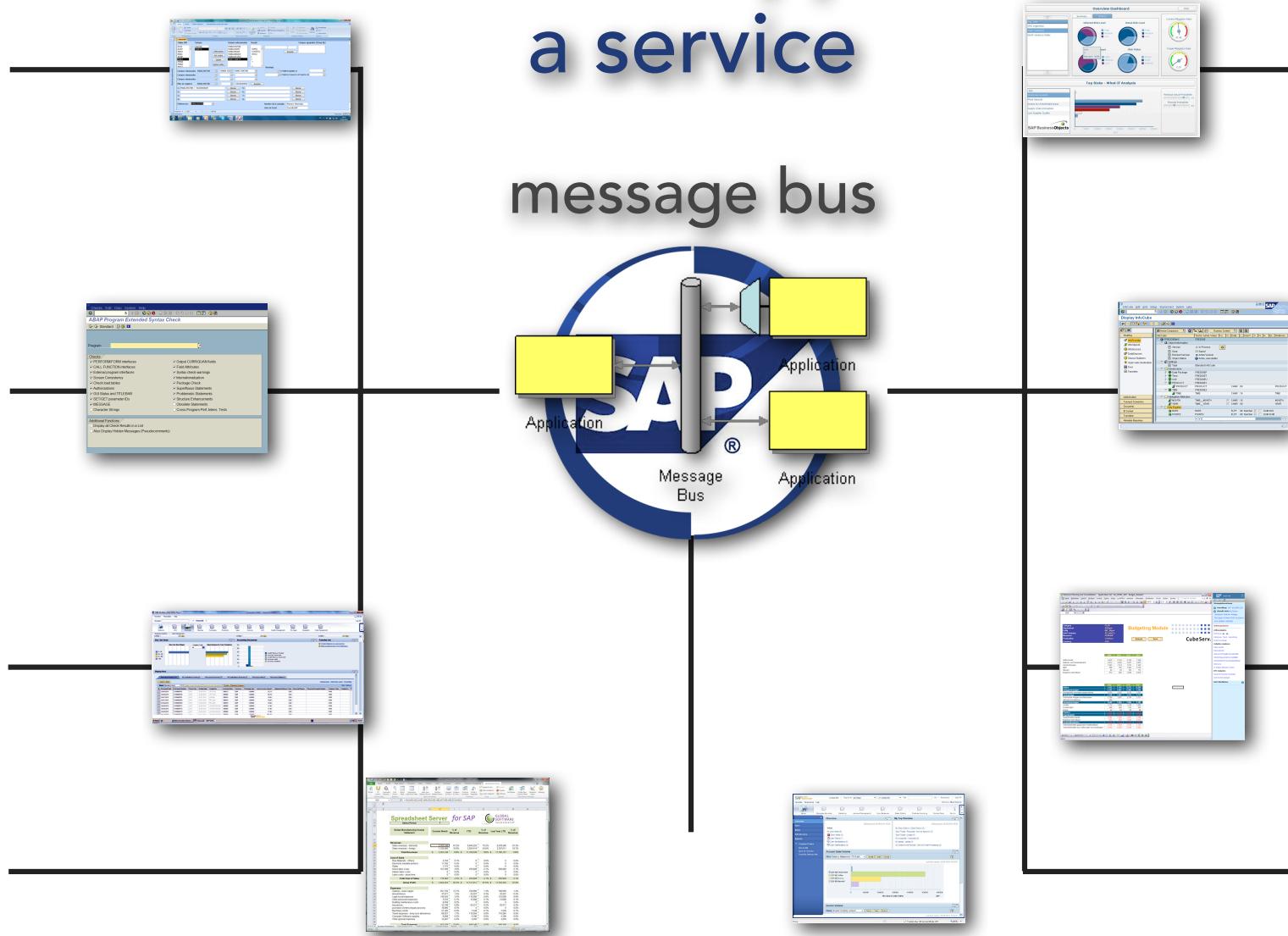


vendor king



vendor king

vendor app as a service



big bang architecture

big bang architecture

designing the entire architecture at the beginning of
the project when you know least about the system

big bang architecture



only architect what is absolutely necessary to get the project started and on the right track

let the architecture evolve throughout the project as you discover and learn more about the system

don't forget - requirements, technology, and business needs change constantly - and so must the architecture

? ' 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: +1 404 4242 9929 Twitter: @neal4d

E: nford@thoughtworks.com W: thoughtworks.com