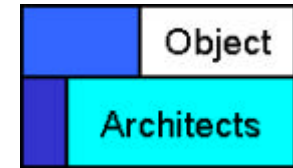


Enterprise Level Architecture Management

Part IV: Mergers and Other Typical Decision Situations

Wolfgang Keller, Plattform-Management,
Generali Office Service & Consulting AG, Wien

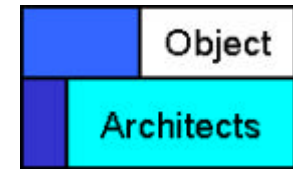
Email: wolfgang.keller@generali.at
<http://www.objectarchitects.de/>



Contents

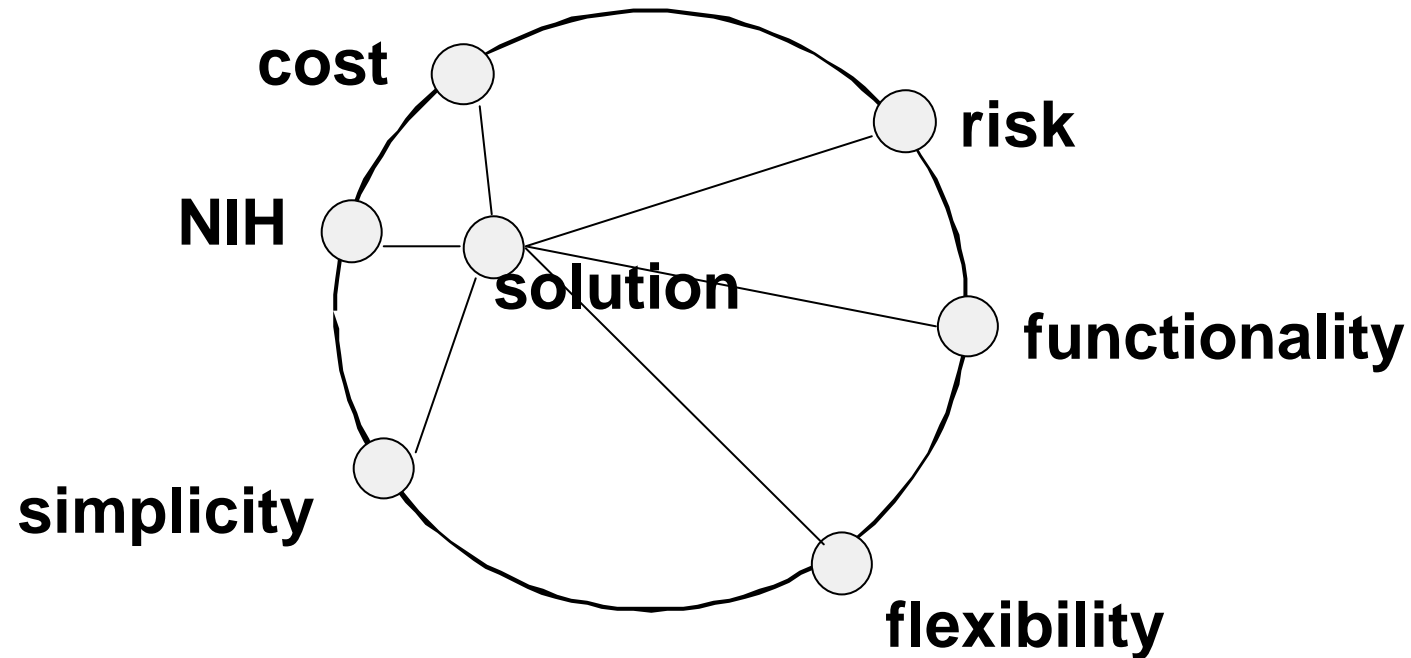
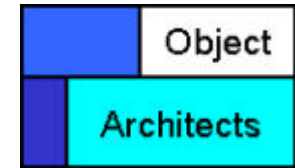
- forces
- a few patterns
 - One Infrastructure / One Application Wins
 - Keep the Data – Toss the Code
 - Early Decision

Forces

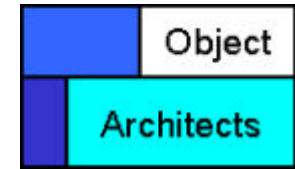


- cost, synergies and economies of scale
- socio cultural forces
 - who's the boss?
 - NIH syndrome
- risk
- functionality
- nonfunctional requirements

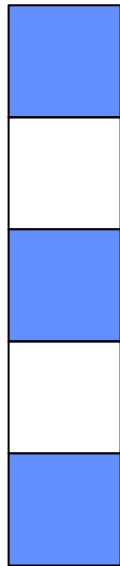
on the nature of forces ...



Why would you want to merge two financial institutions



5%+



Small insurance
<~ 1 billion Euro

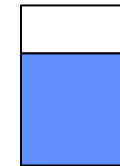
typical edp cost / premium ratios

2,5 – 3,5%+



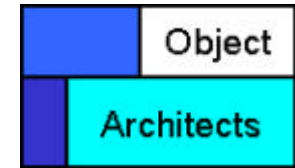
medium insurance
~ 5 billion Euro

1,5 – 2%



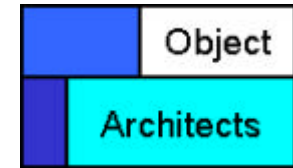
Large insurance
~20 billion Euro

why would you want to merge two financial institutions



- multi-channel approaches open multiple new channels to spend more money on the same customers
- Low costs allow you to offer the same products for a lower price
- There are significant economies of scale in the financial industry ..

there are significant economies of scale in the financial industry ..



large
German bank

~ 1200
employees
in edp

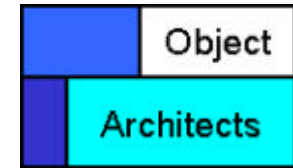
+ „large“
Austrian bank

~ 1200
employees
in edp

merged bank

~ 1200+x
employees
in edp

socio-cultural forces who's the boss?

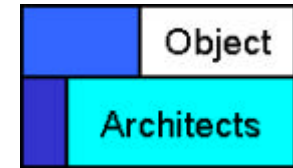


- Bank A „buys“ Bank B in a „merger of equals“
- Bank A's managers are slightly more equal than Bank B's

- which system will be chosen?
 - => depends
- who will head the joined edp department
 - => make a guess ✍

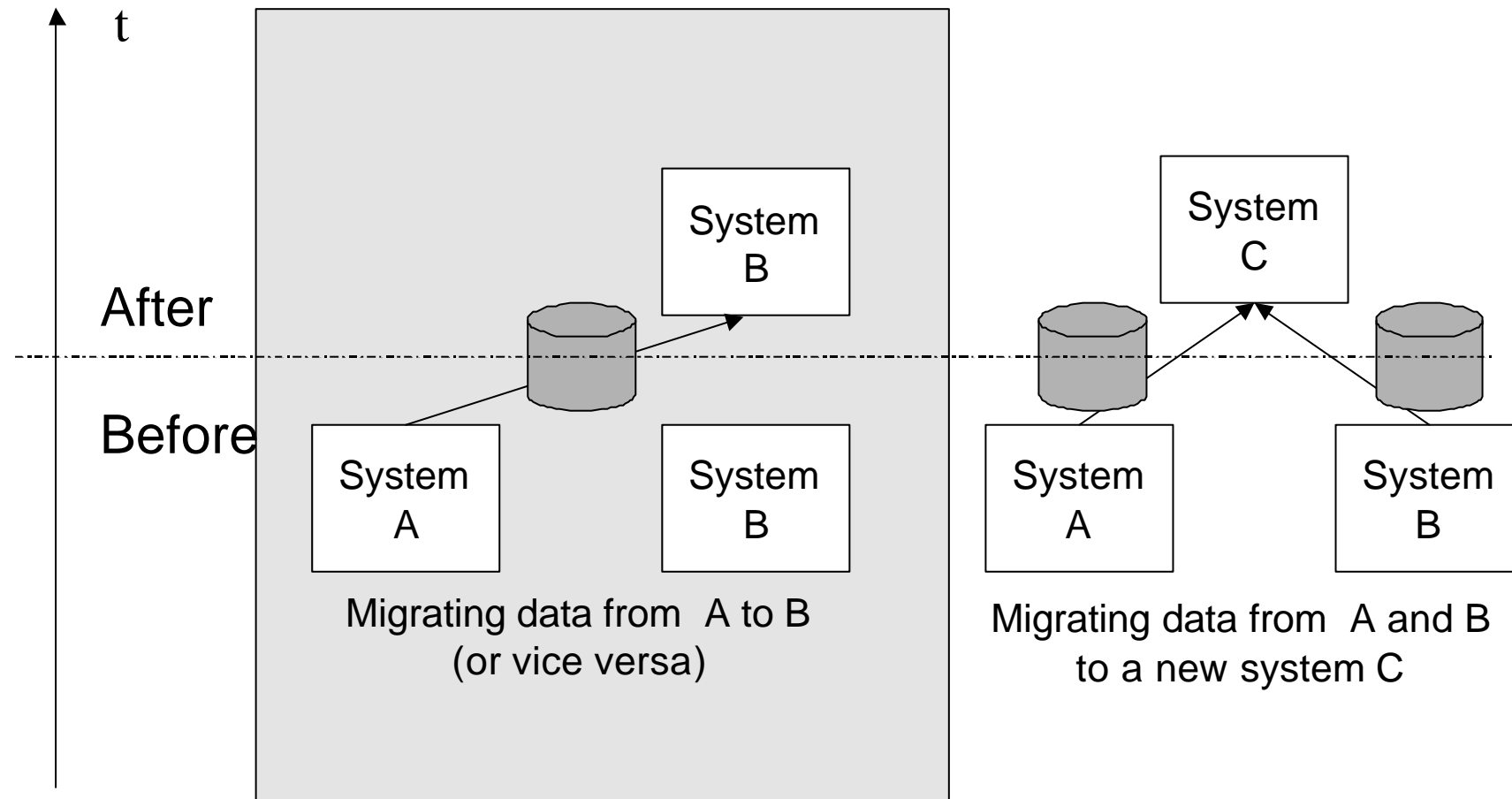
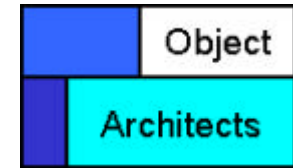
socio-cultural forces

NIH syndrome

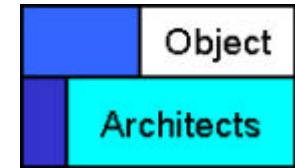


- whose solution will a developer prefer?
- his own solution?
- another developers solution?
 - that threatens to put him out of work
 - that will put him out of charge
 - that means that he has to learn a lot of new things ..

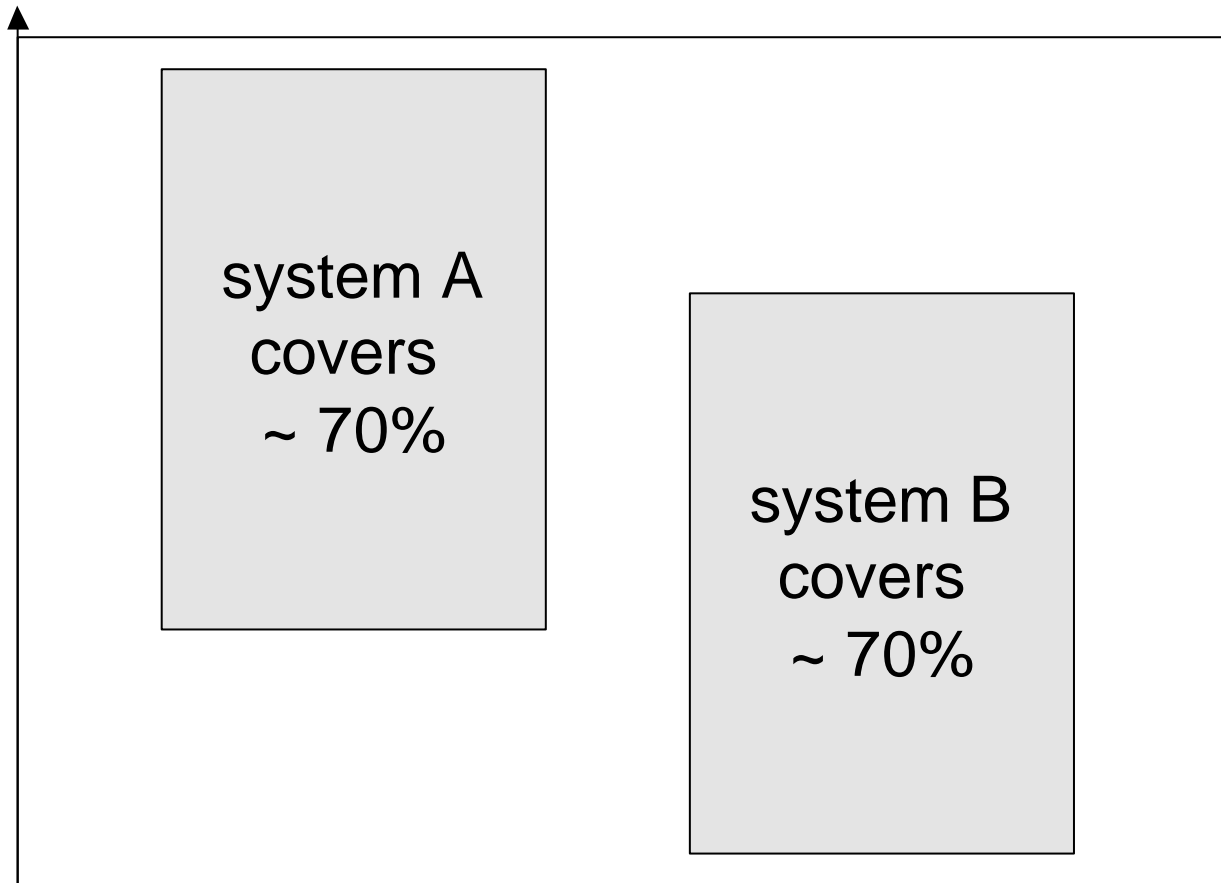
risk: which migration scenario has lower risk?



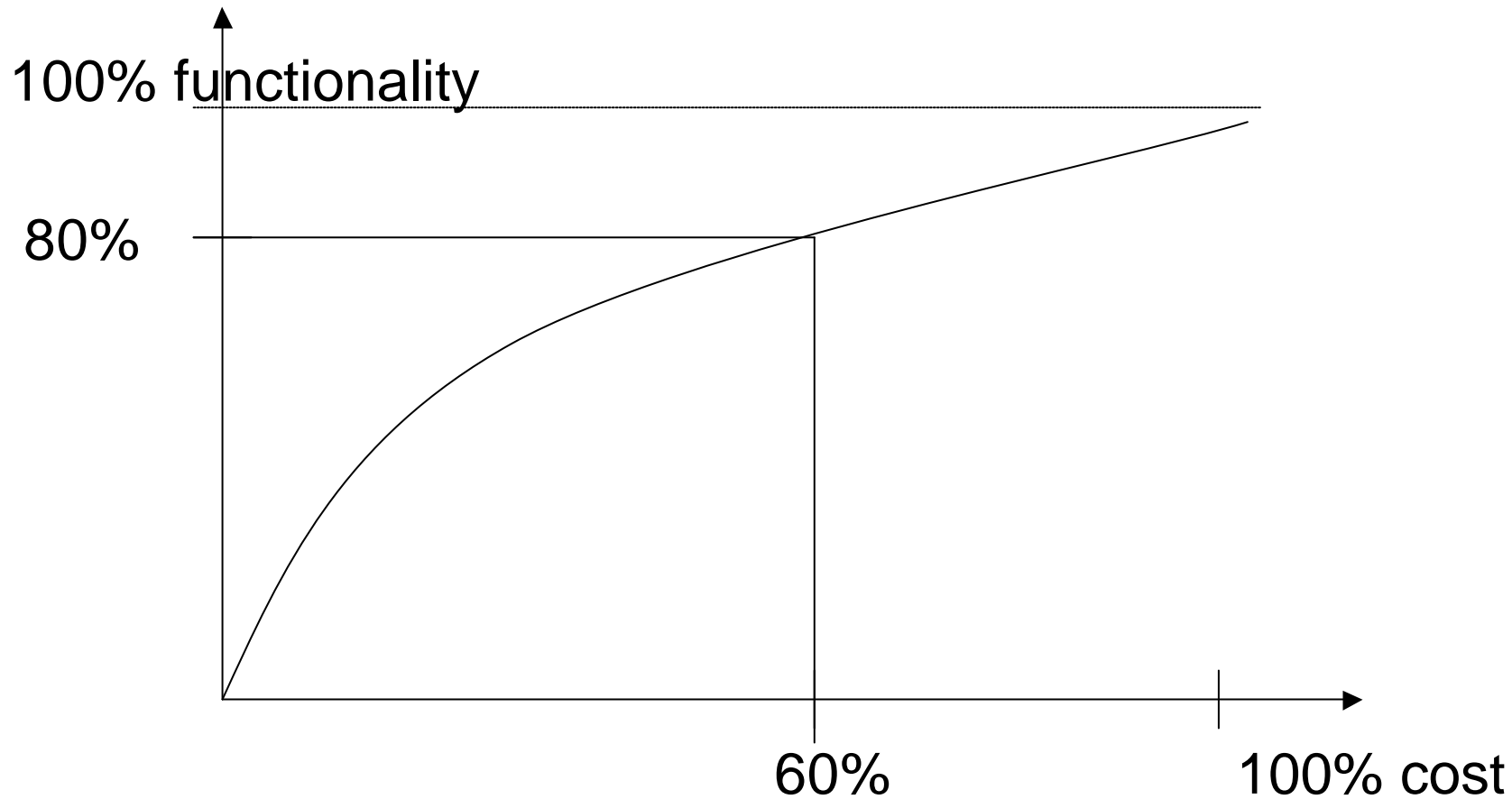
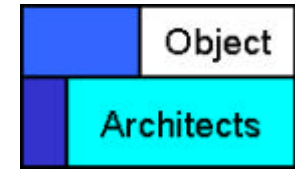
functionality: compare two established systems

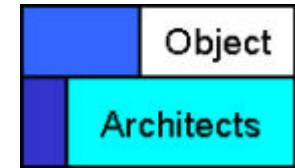


100% possible functionality



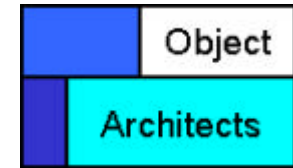
80/20 principle a.k.a. good enough





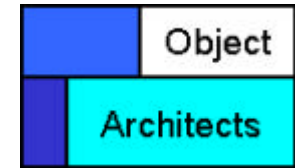
The last two slides imply ...

- often it does not really matter which of two established systems you choose – at least from a standpoint of functionality ...
- never try to merge a 100% solution from two 70% solutions using code level reengineering ...
- never try to build a system with 100% functionality – it's too expensive



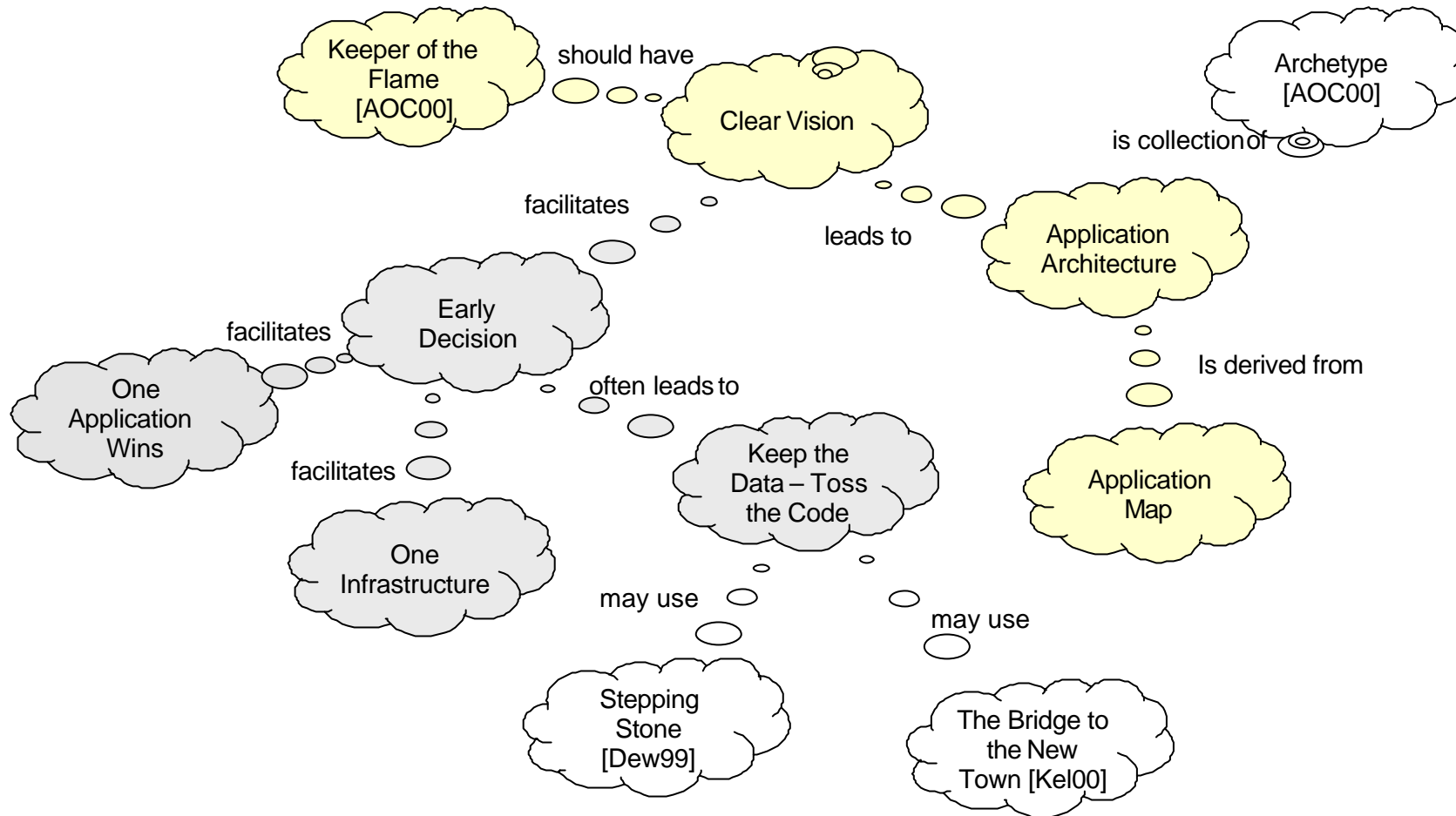
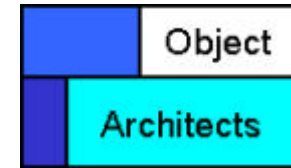
and finally ...

- there's also technology and nonfunctional requirements such as ...
 - technical quality
 - maintainability
 - ...
 - ..
 - .
 - object-orientation

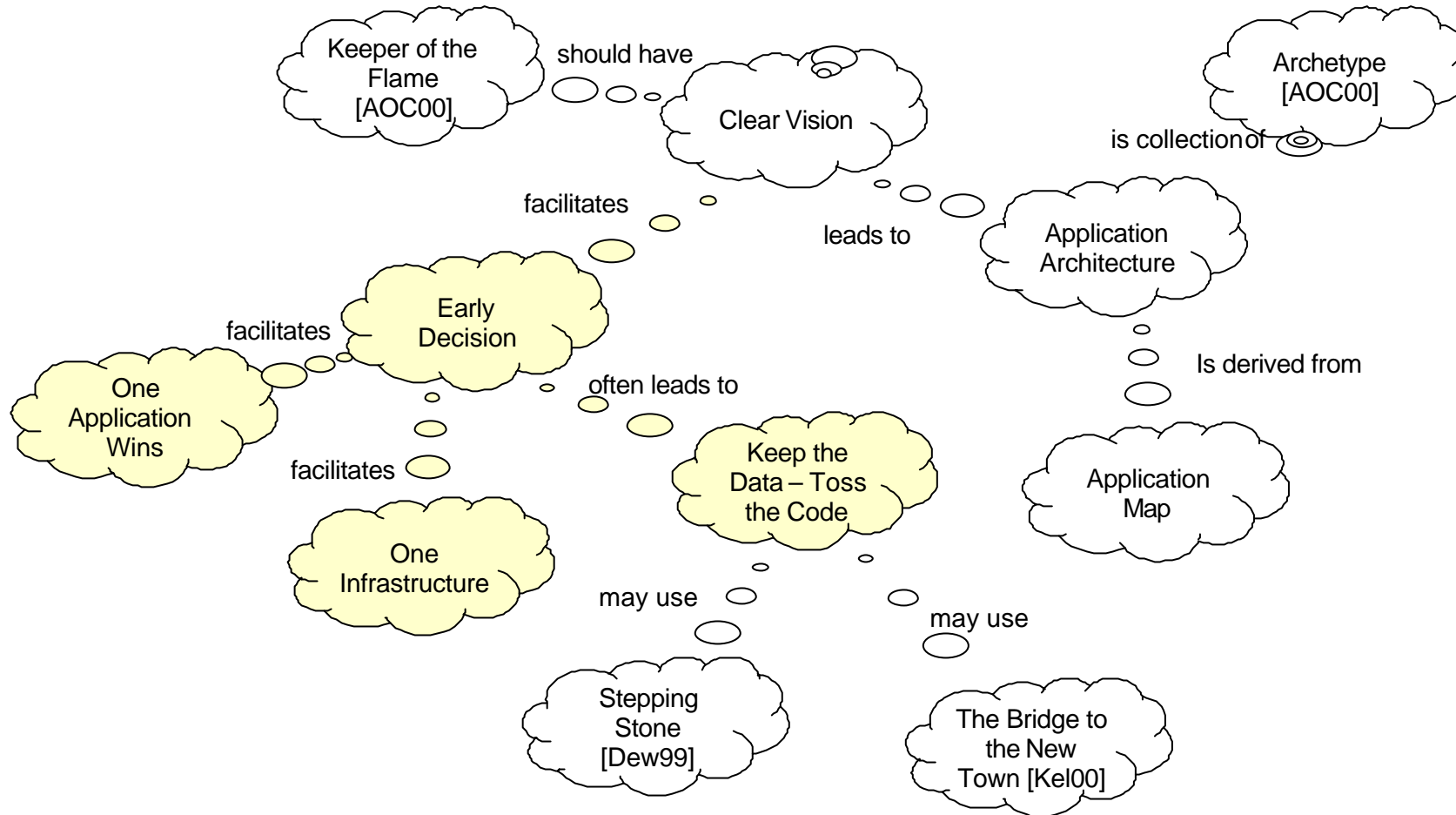
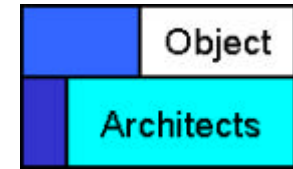


a few patterns

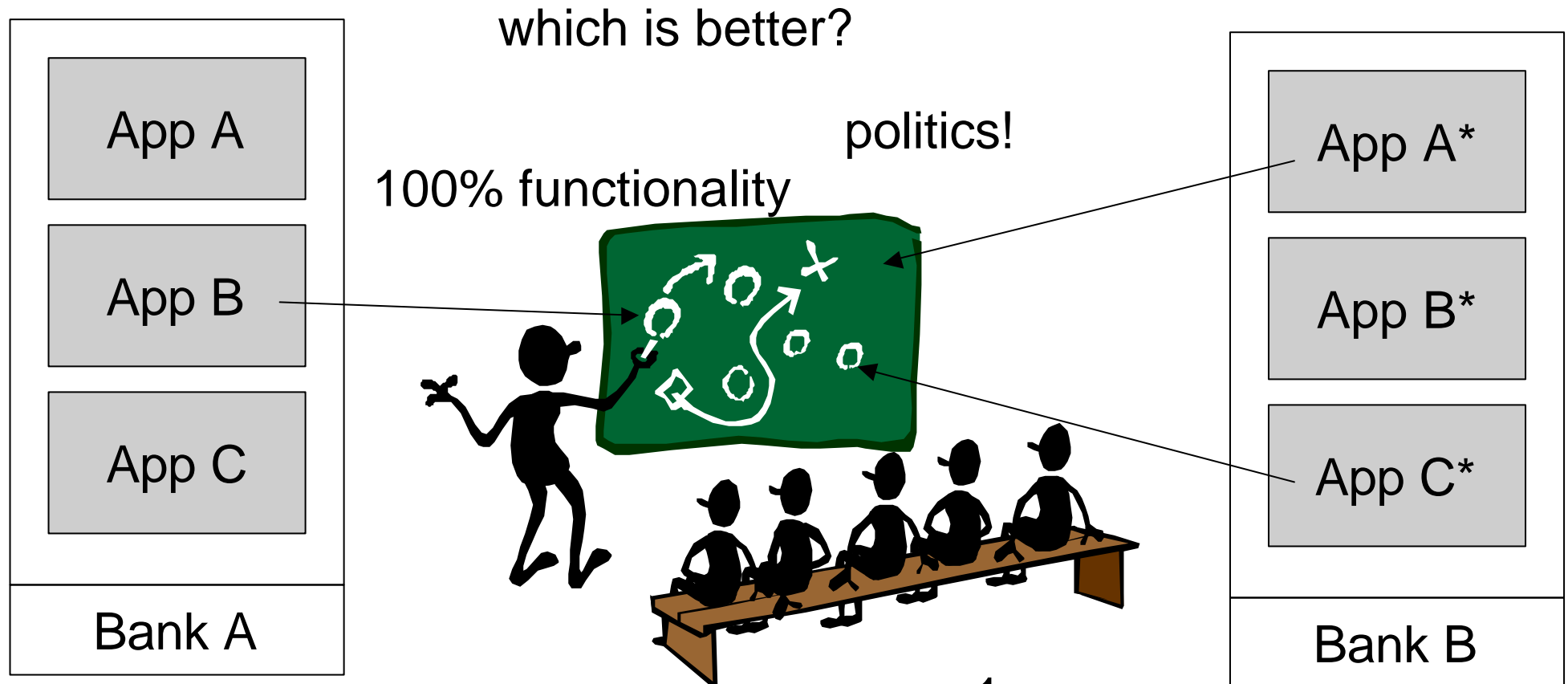
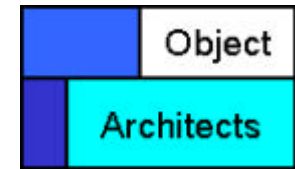
a few important patterns ... from the architecture part ...



and the ones we will discuss in this part ...



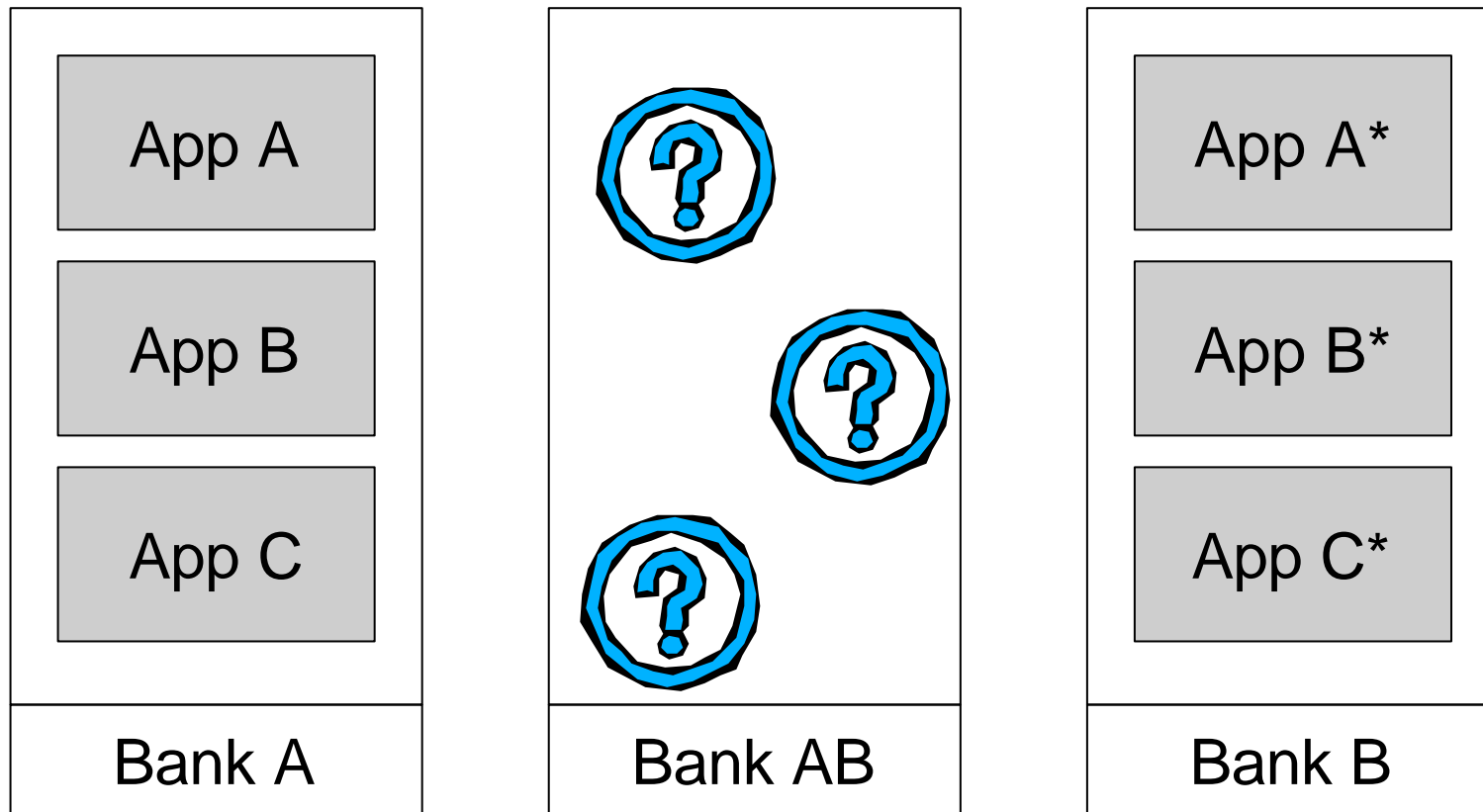
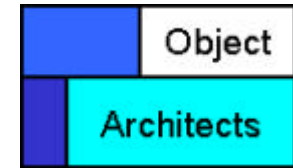
after the merger a series of workshops starts ...



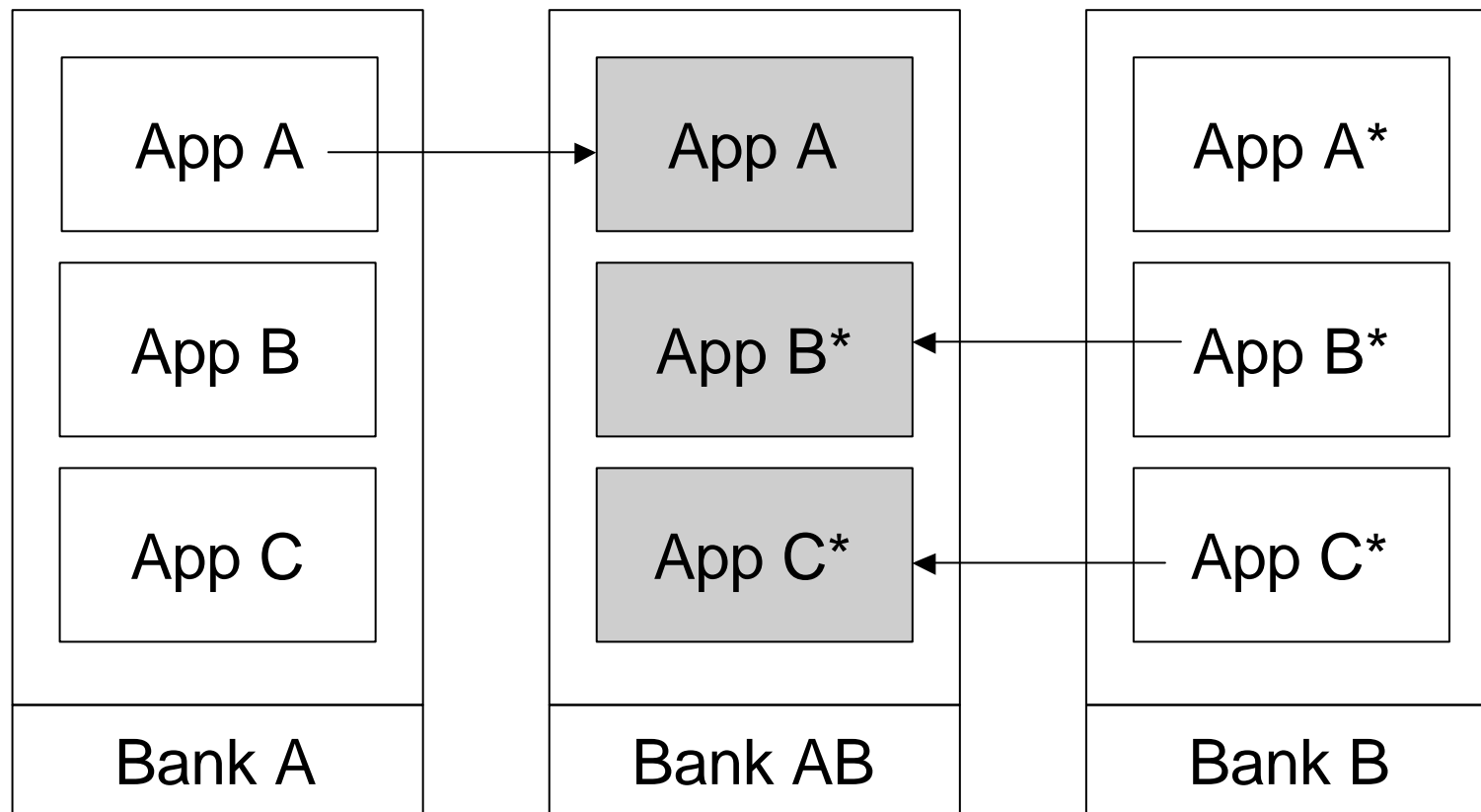
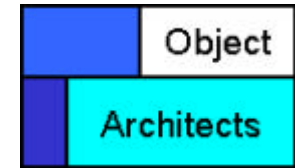
NIH syndrome

1 year
decision process

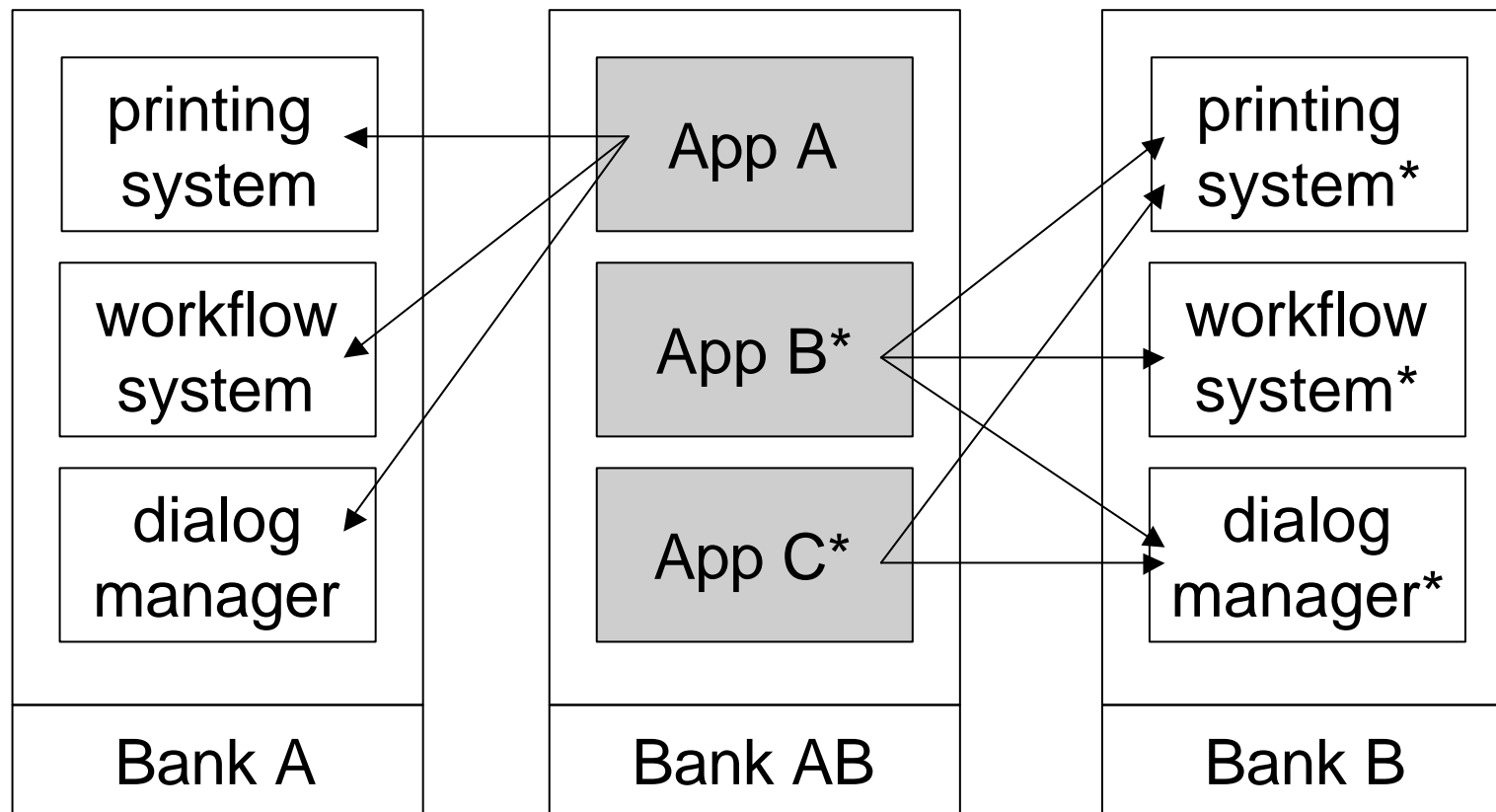
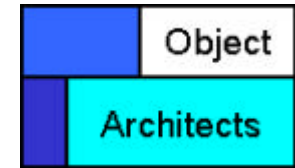
pattern: one infrastructure pattern: one application wins

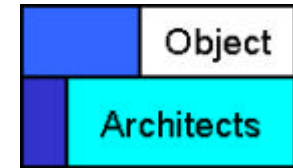


one might think it is a good idea to use the best from Bank A and Bank B ...



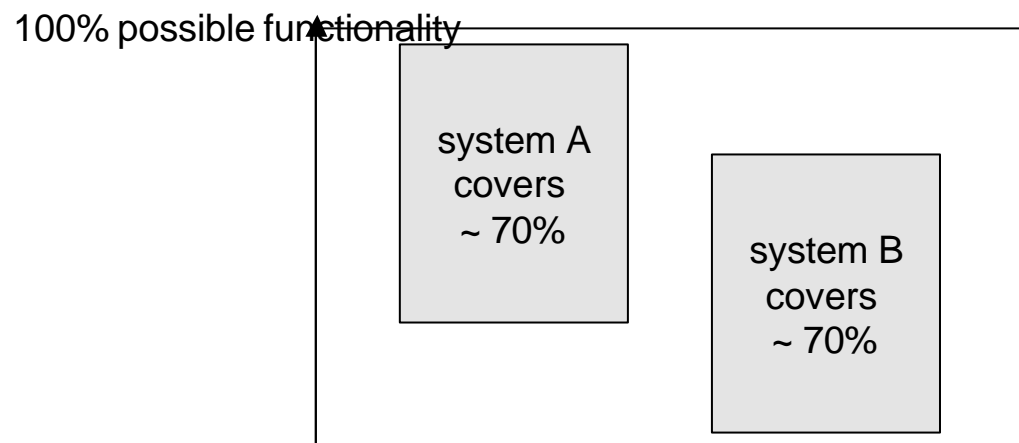
but .. each system uses infrastructure like workflow, dialog managers, printing ...



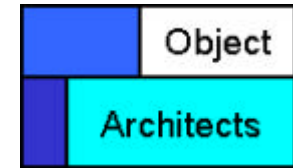


therefore ..

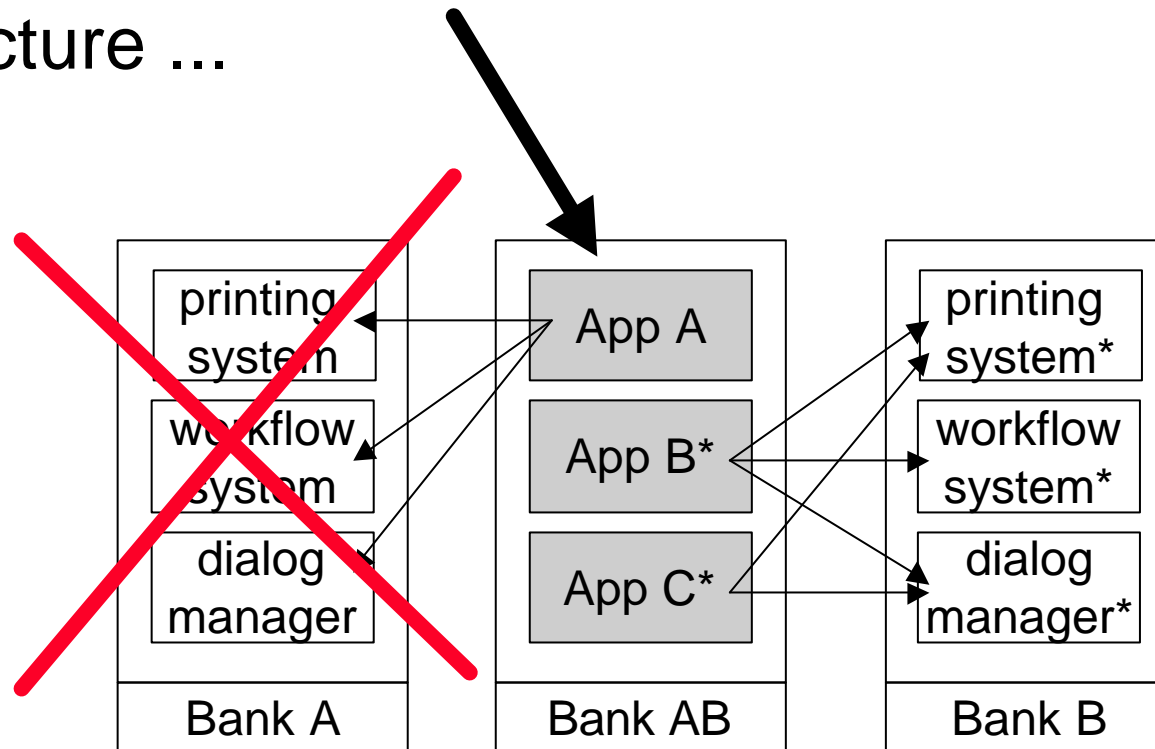
- try to avoid a mixed portfolio
- if the two systems are well established and support the business in a sufficient manner, functionality is not really a question – because it does not matter, which 70% you have ..



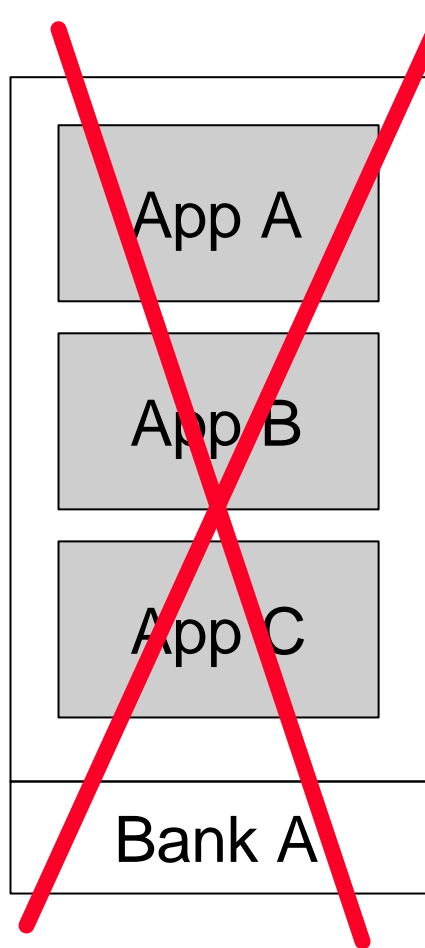
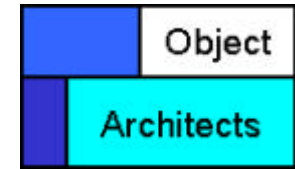
other downsides of the „mixed approach“ are



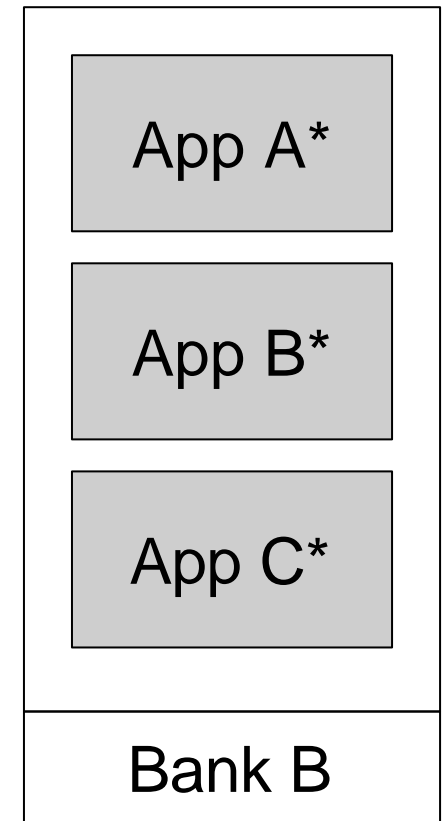
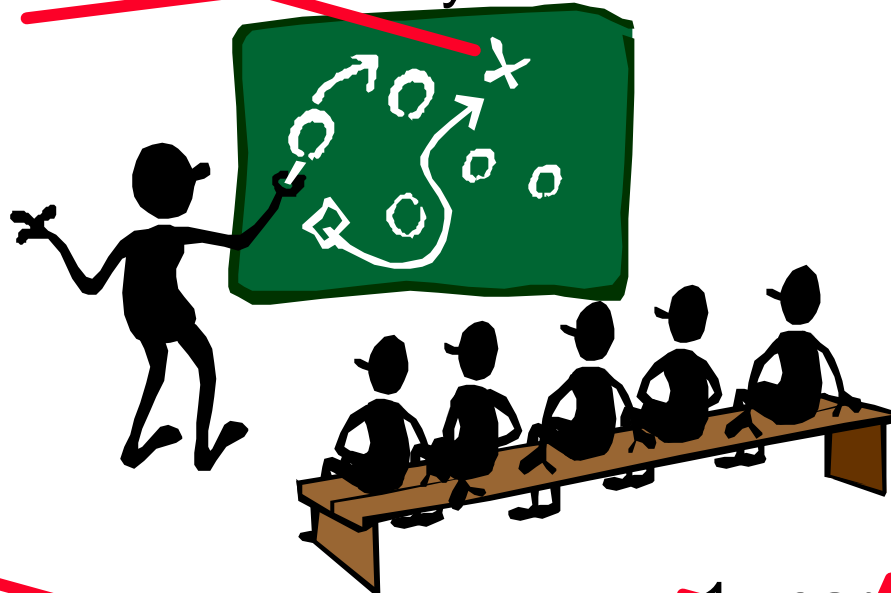
- you need code level migrations to kick out one set of infrastructure ...



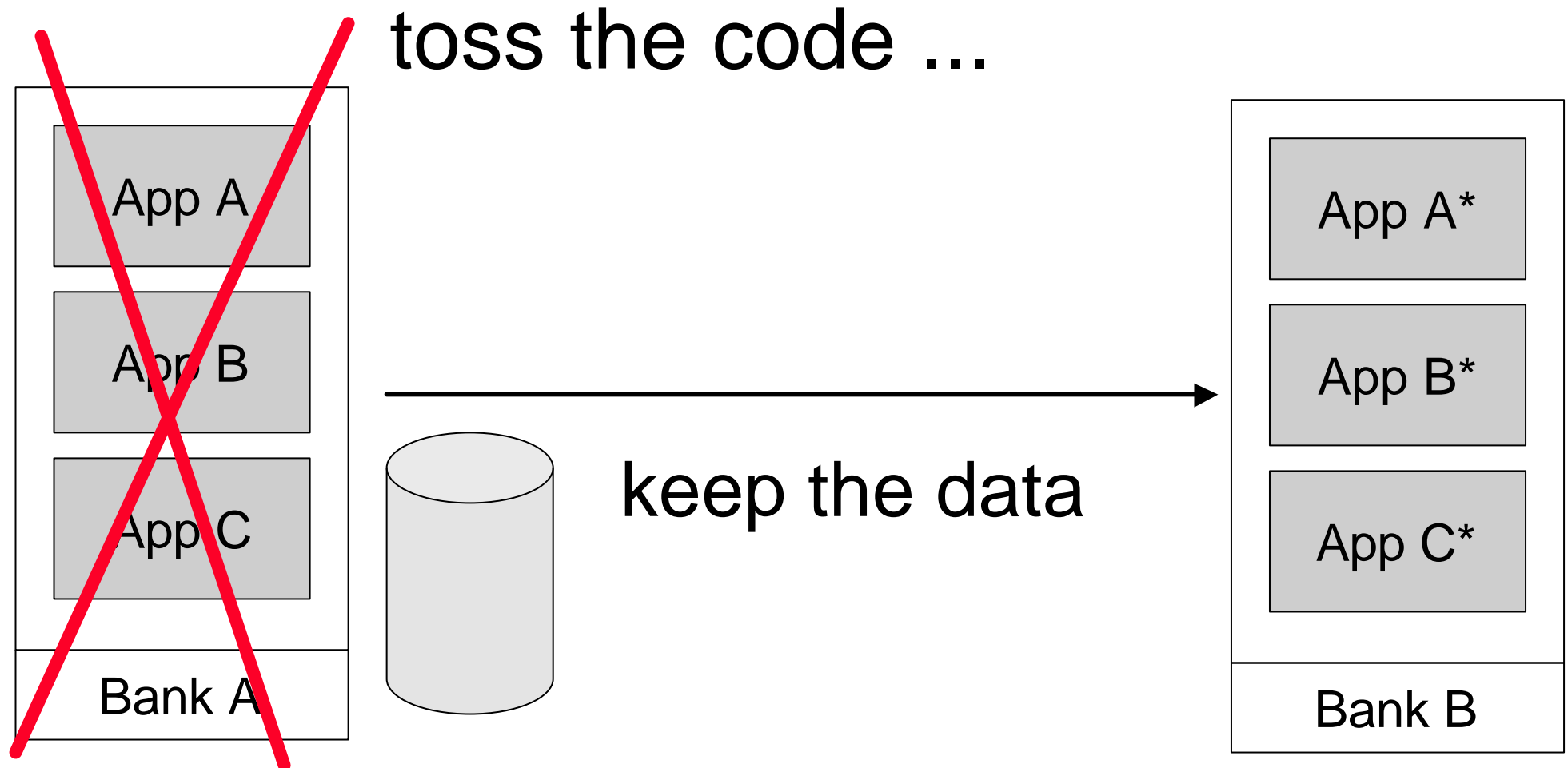
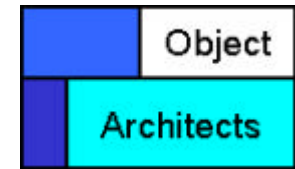
pattern: early decision decide fast and hard ...



which is better?
~~100% functionality~~ ~~politics!~~



pattern: data migration



disclaimers

- pure radical solutions are seldom really good
- 1 year of workshops might not add value – no workshops at all might lead to a fatally bad decision and might also accumulate unnecessary political resistance

