THE UNIVERSITY OF BRITISH COLUMBIA

Flooding and Recycling Authorizations

Konstantin (Kosta) Beznosov

Laboratory for Education and Research in Secure Systems Engineering

lersse.ece.ubc.ca

outline

- the problem
  - assumptions
  - target environments
  - limitations of point-to-point architectures
- the approach
- summary & future work

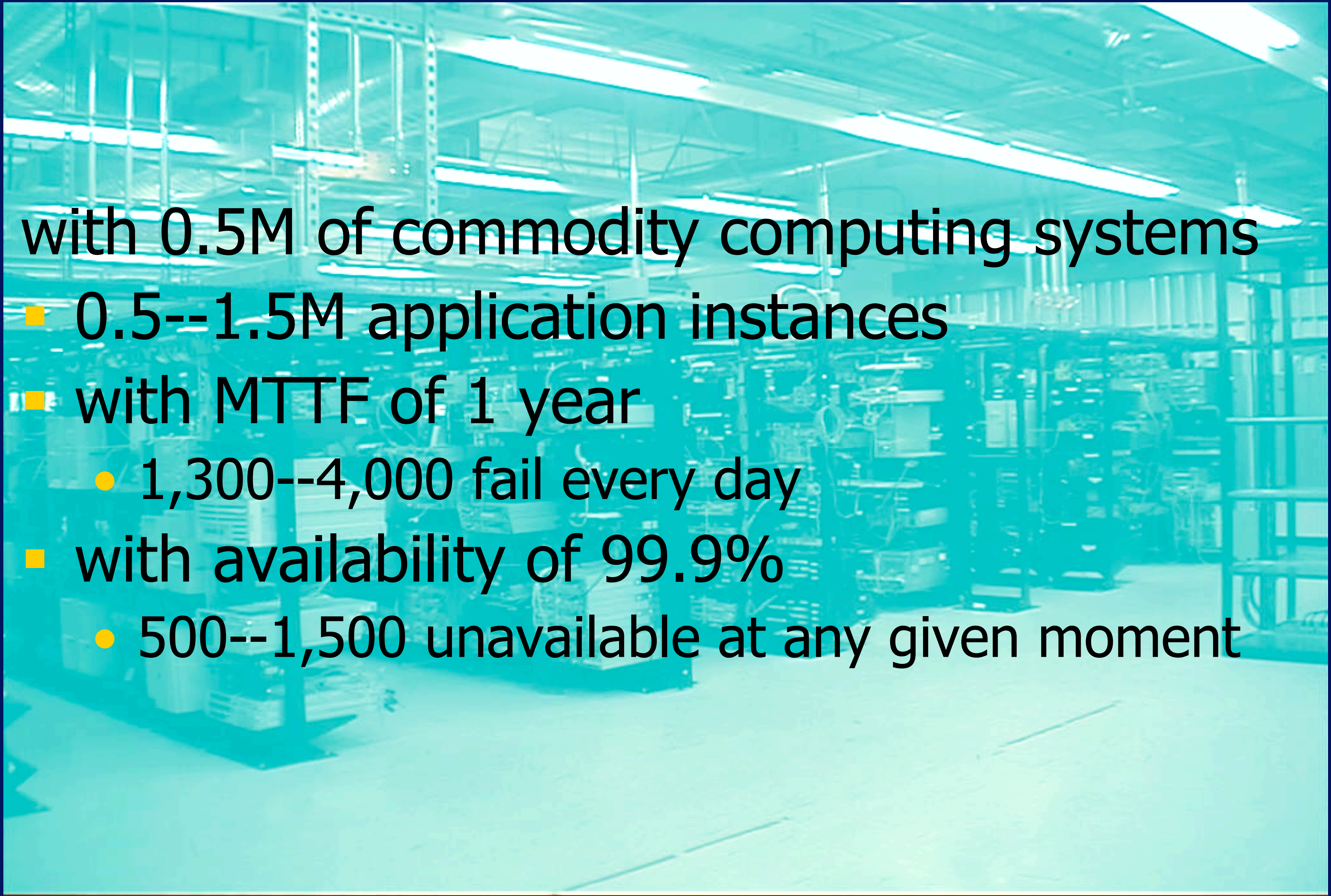
the problem

departing assumptions

- processor resources virtually free
- commodity computing most cost-effective
- network bandwidth virtually unlimited
- human time/attention expensive

target environments

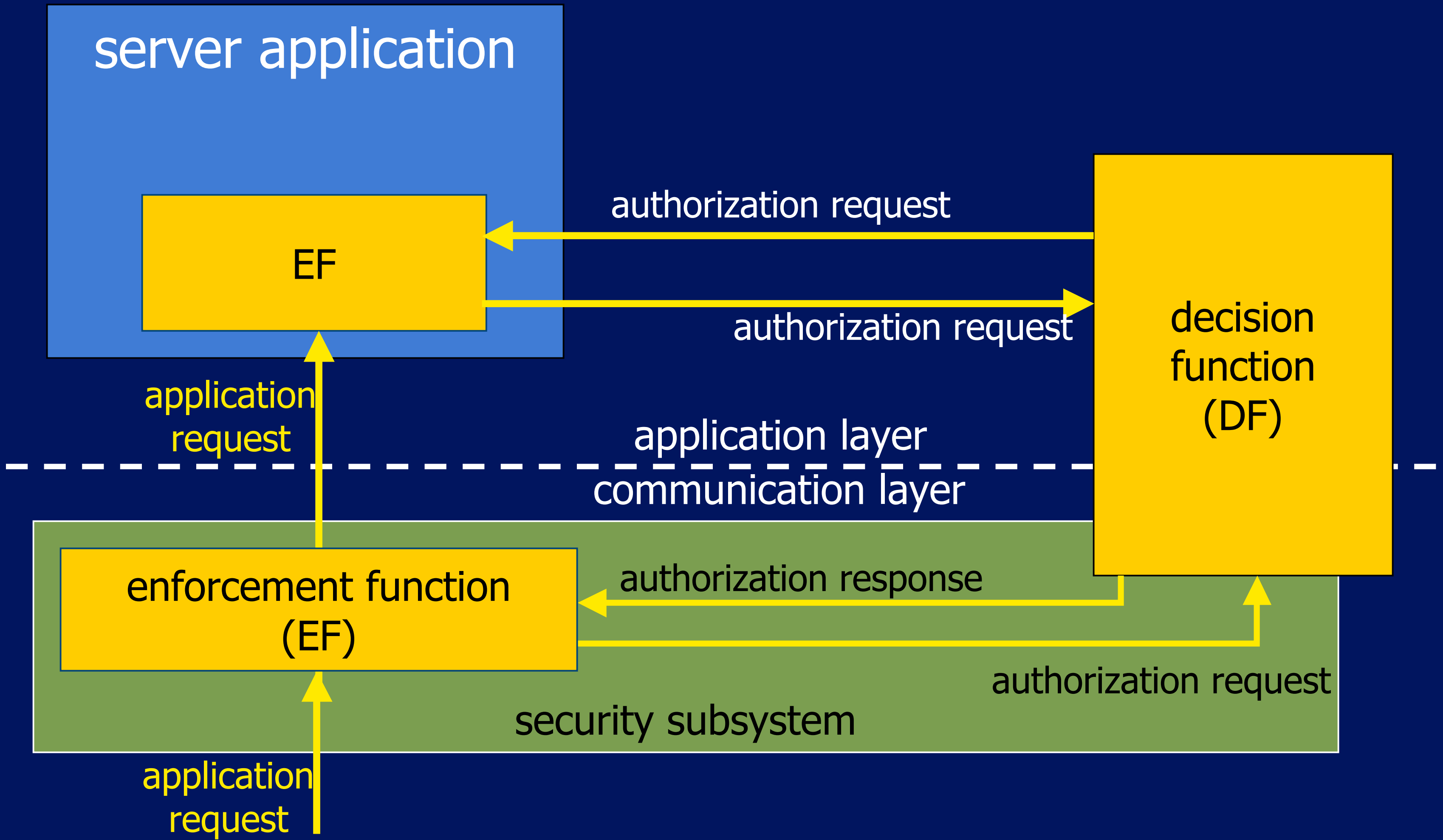
target environments



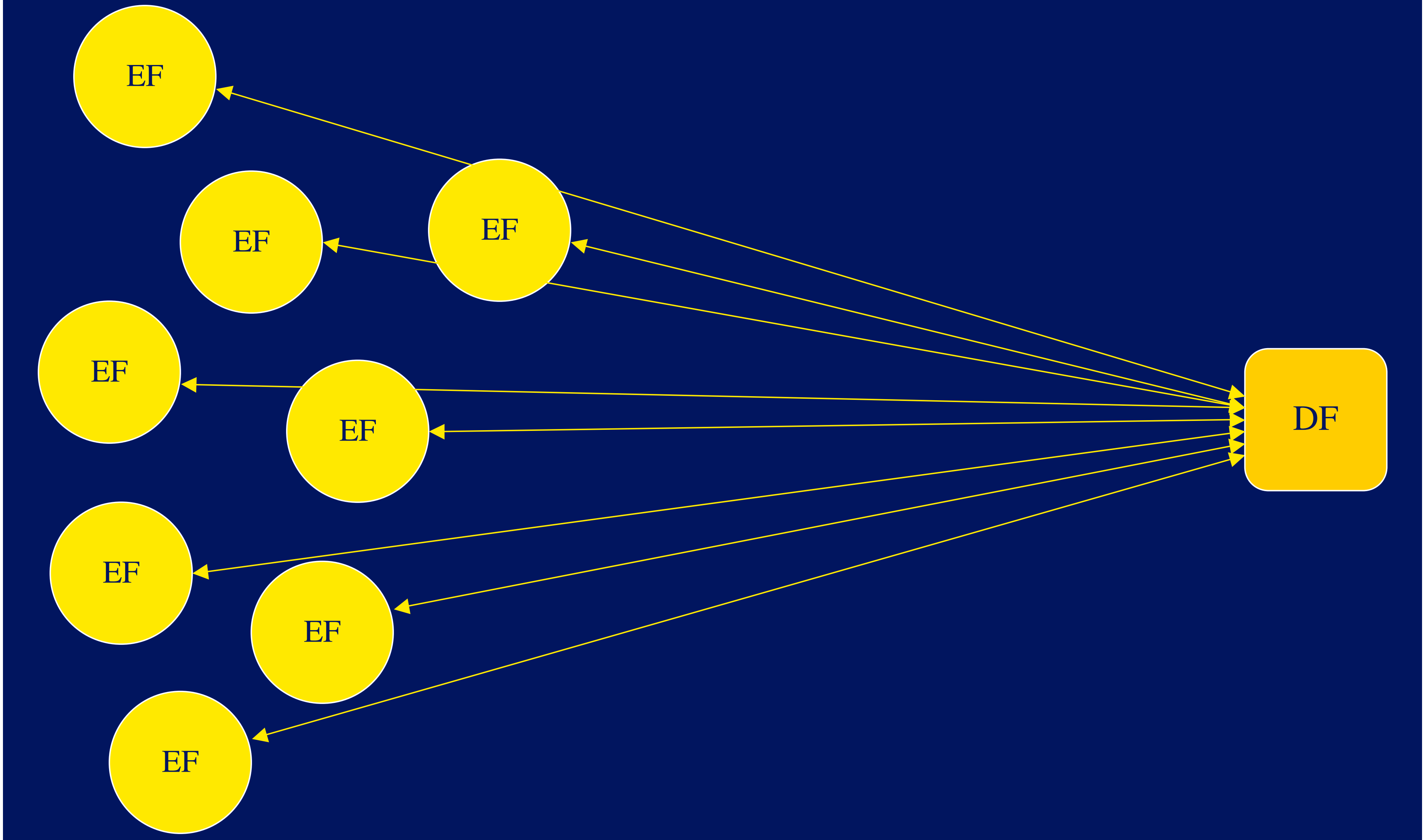
with 0.5M of commodity computing systems

- 0.5--1.5M application instances
- with MTTF of 1 year
  - 1,300--4,000 fail every day
- with availability of 99.9%
  - 500--1,500 unavailable at any given moment

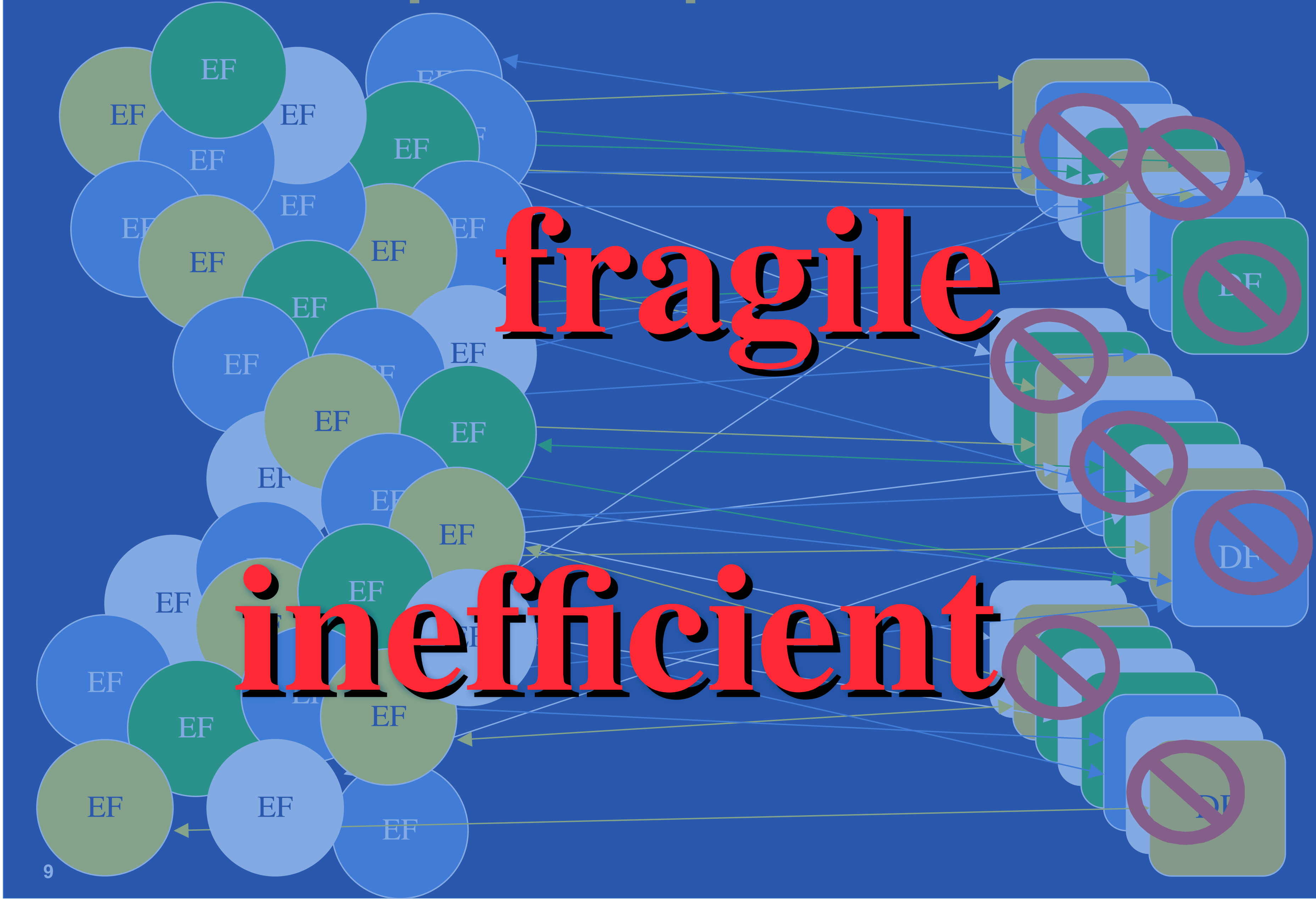
request-response paradigm



enables DF reuse



results in point-to-point architectures



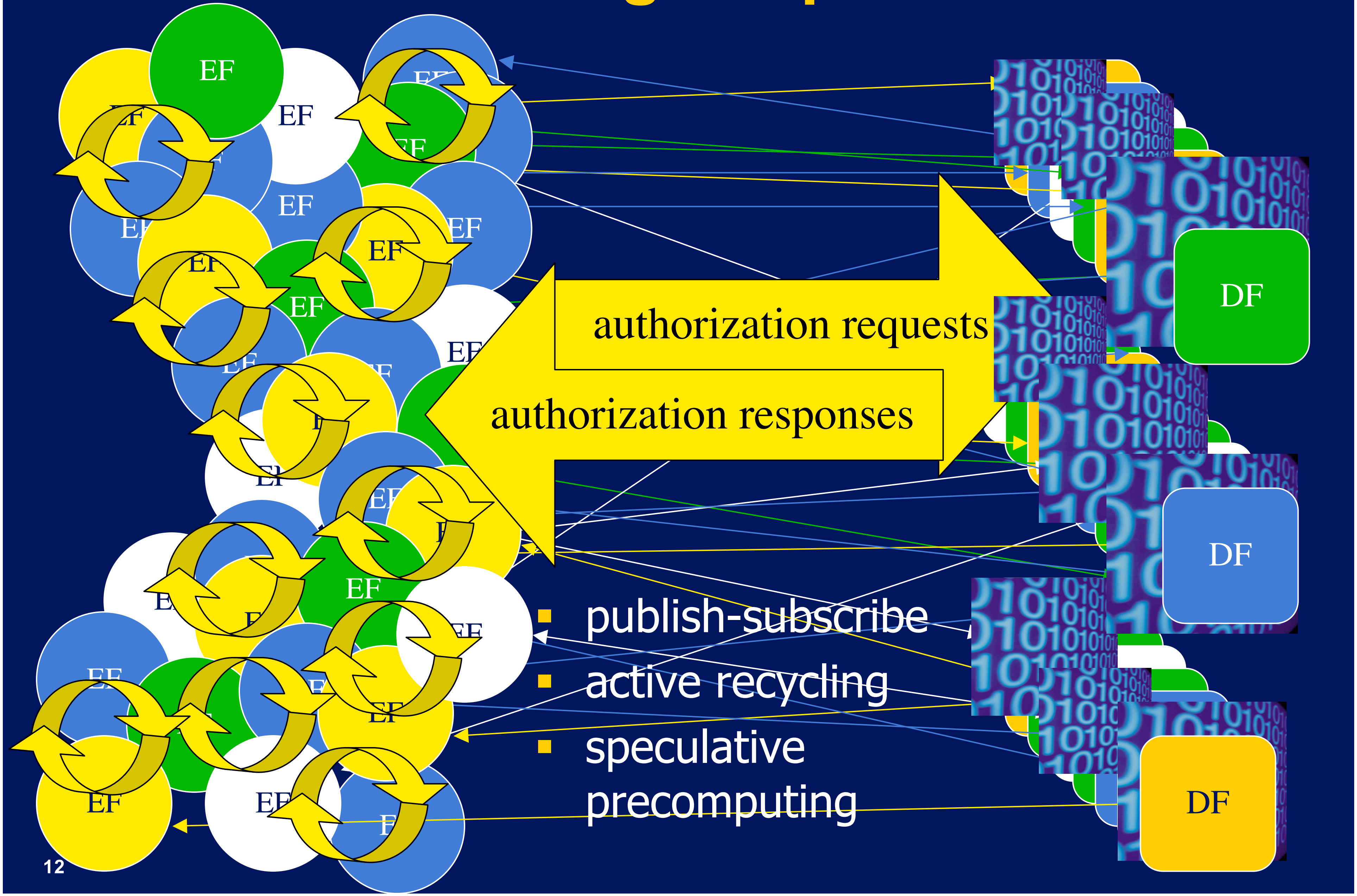
addressed problem

point-to-point authorization architectures at massive scale

- become too fragile
  - require costly human attention
  - jeopardize organizational goals
- fail to reduce latency
  - security-related performance overhead too high

proposed approach

addressing the problem

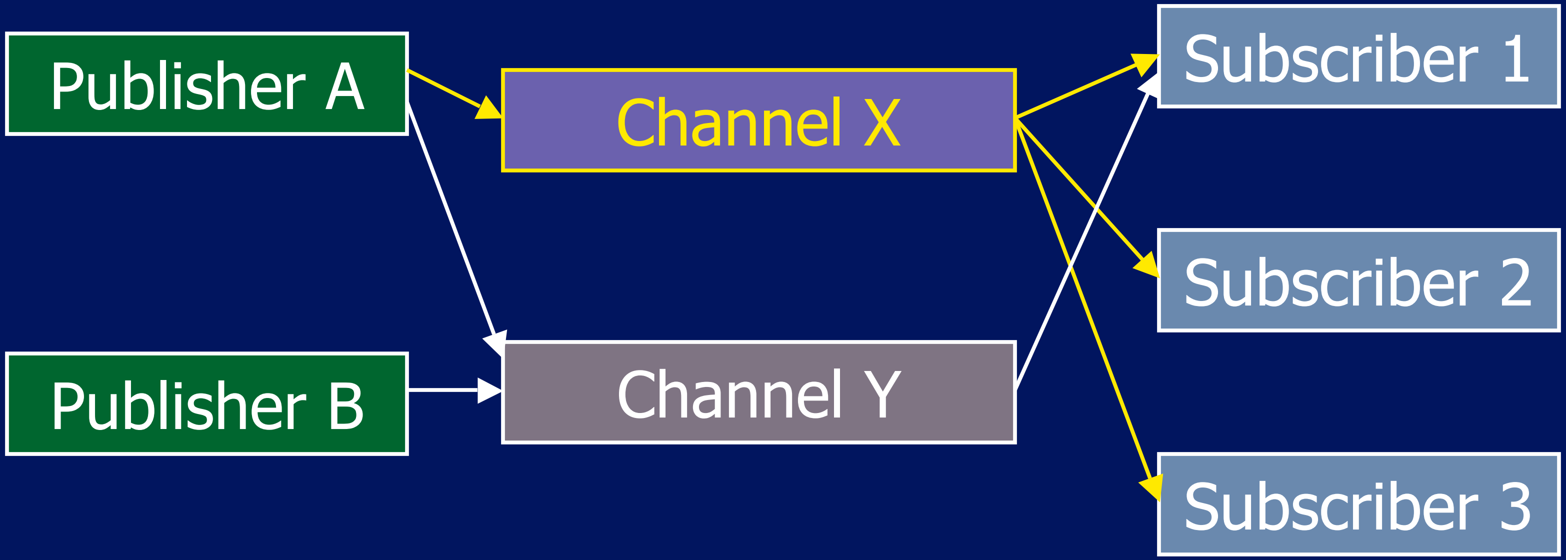


- publish-subscribe
- active recycling
- speculative precomputing

THE UNIVERSITY OF BRITISH COLUMBIA

publish-subscribe

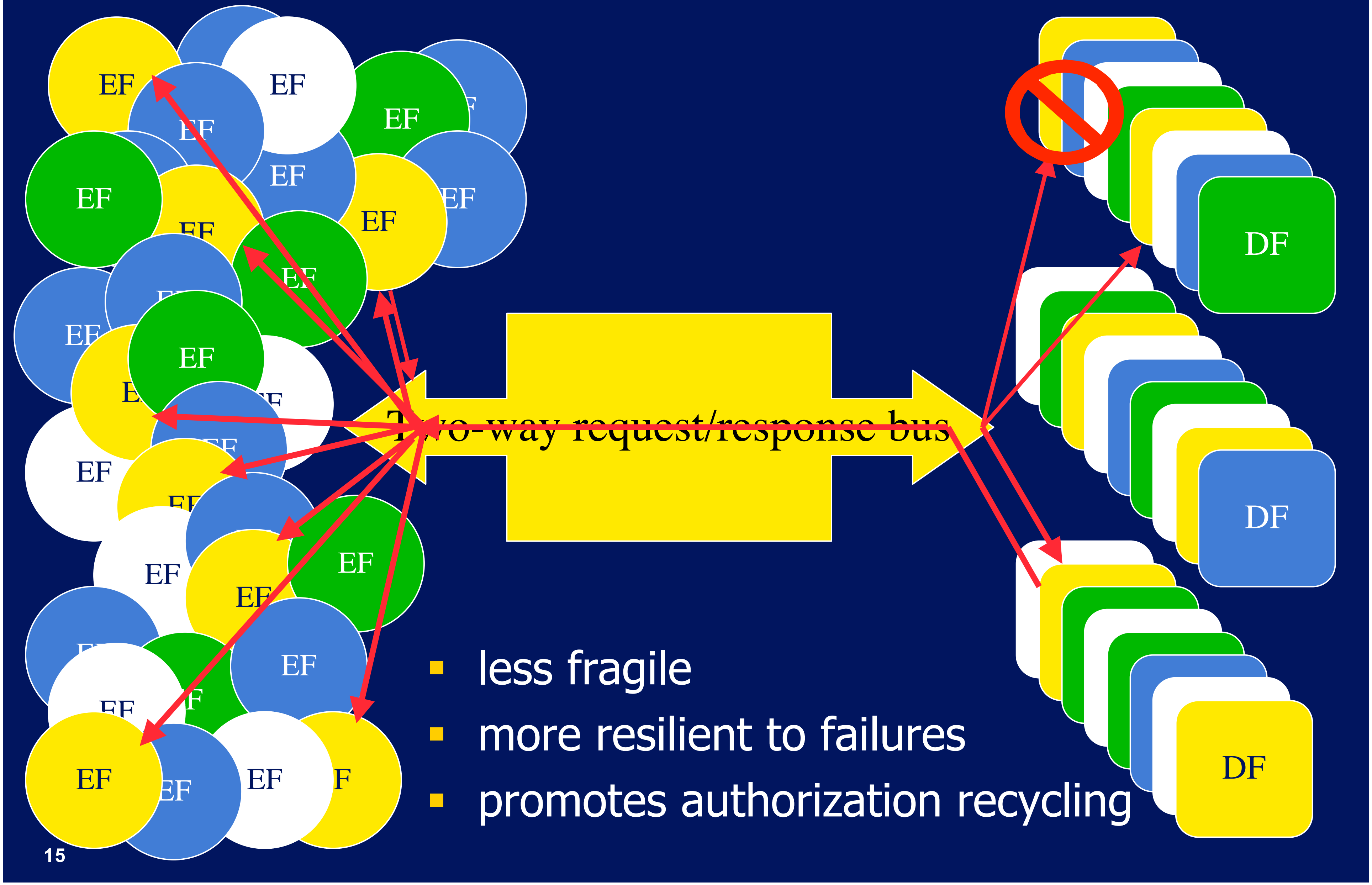
publish-subscribe architecture



Used properties:

- many-to-many
- asynchronous

publish-subscribe for policy decisions

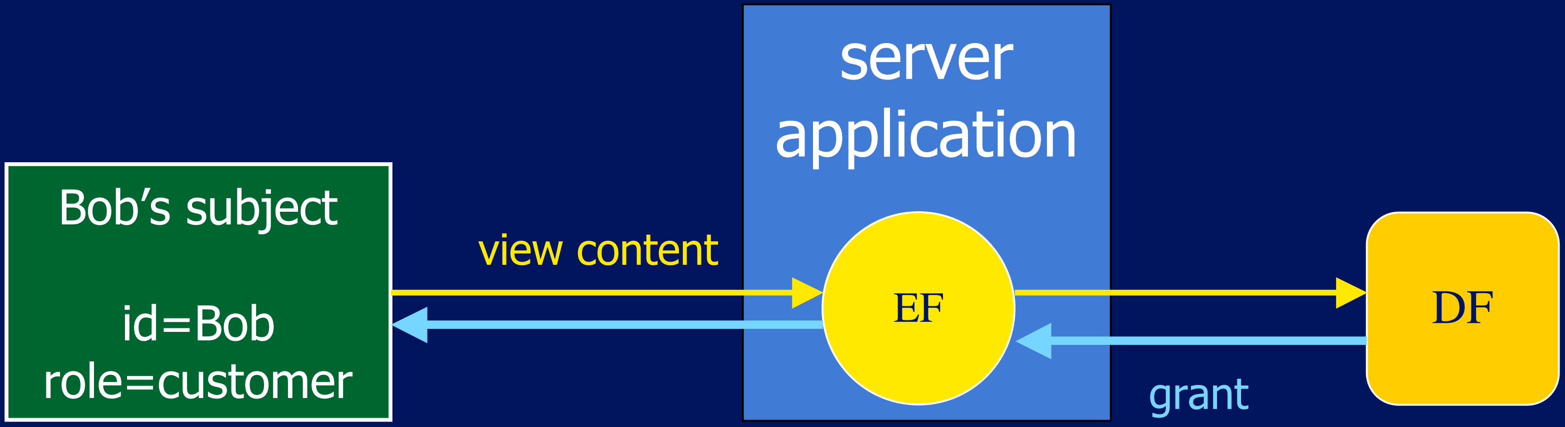


- less fragile
- more resilient to failures
- promotes authorization recycling

THE UNIVERSITY OF BRITISH COLUMBIA

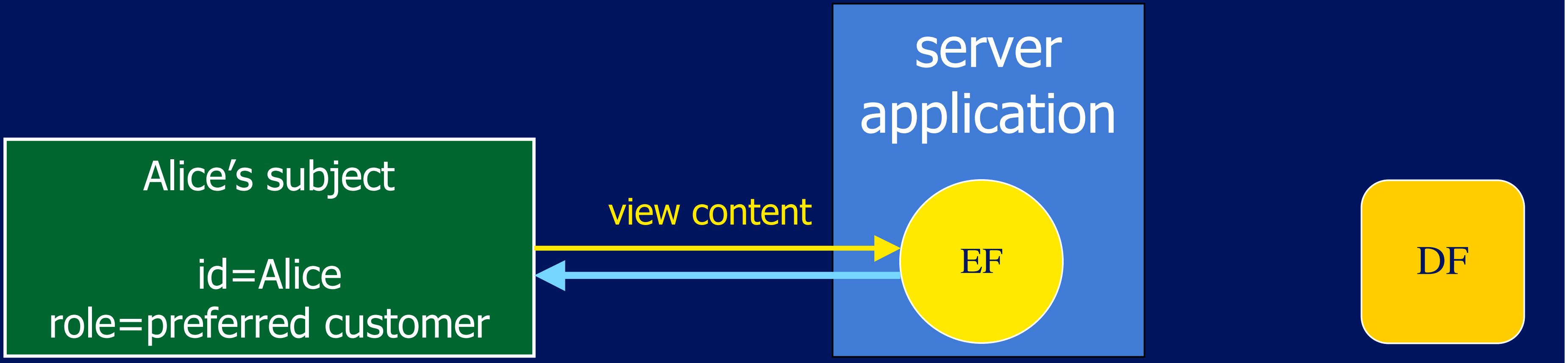
active recycling of authorizations

intuition



Authorize Bob to view content for customers

intuition



Alice's subject

id=Alice
role=preferred customer

view content

server application

EF

DF

Authorize
Alice to view
content for
customers

model
in progress

basic elements

- request

  <subject, object, access right, context, request id>

  < s , o , a , c , i >

  <{id="Bob", role="customer"}, {id="eB-23"}, view, {date="05-08-15"}, 6112>

- response

  <response id, request id, Evidence, decision>

  < r , I , E , d >

  < 934598438, 6112, [ ], allow > -- direct (from DF) response

  <{id="Bob", role="customer"}, {id="eB-23"}, view, {date="05-08-15"}, 6115>

  < 943498843, 6115, [934598438], allow > -- indirect/precise response

  <{id="Alice", role="pr. cust."}, {id="eB-23"}, view, {date="05-08-15"}, 6120>

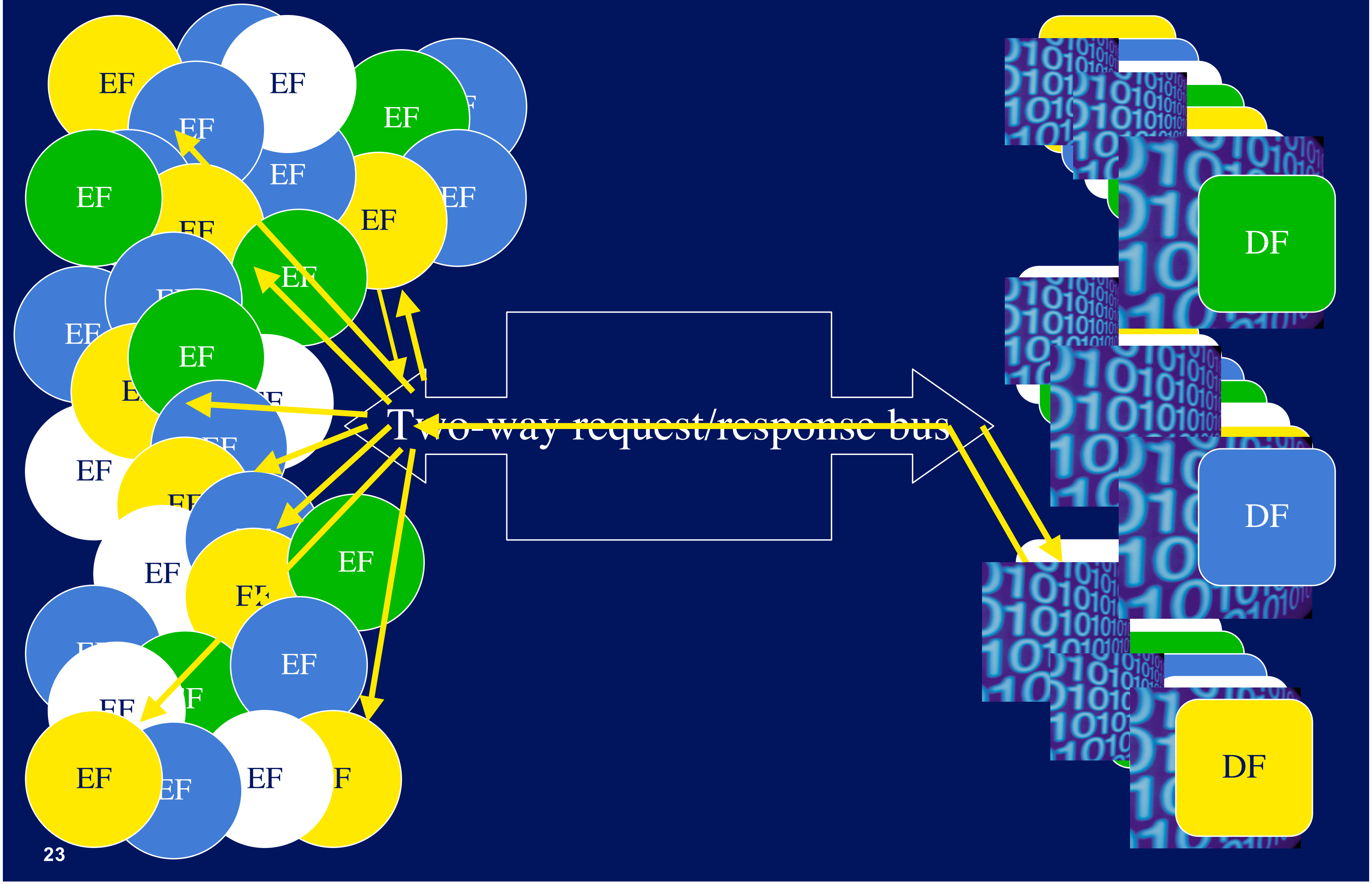
  < 990923124, 6120, [934598438], allow > -- indirect/approximate response

recycling authorizations

- secondary authorizations
  - re-using decisions made for other, but equivalent, requests
  - example <s, o, a, c, i> <s, o, a, c, i’>
- approximate authorizations
  - re-using decisions made for other, but similar, requests
  - examples
    - preferred customer ≥ customer ≥ visitor
    - row ≤ table
    - read ≤ modify

flooding with “junk” authorizations

flooding with speculative authorizations



summary

- problem
  - context and assumptions
    - CPU resources are virtually free
    - commodity computing is most cost effective
    - bandwidth is unlimited
    - human time/attention is too expensive
  - target environments
    - massive-scale enterprises with 10^5 machines
  - limitations of point-to-point architectures
    - too fragile, high latency, too expensive to maintain
- approach
  - decouple EFs and DFs with publish-subscribe
  - recycle authorizations
  - flood with junk authorizations

current status and future work

- current work
  - Secondary and Approximate Authorizations Model (SAAM)
    - $\mathrm{SAAM}_{\mathrm{BLP}}$, $\mathrm{SAAM}_{\mathrm{RBAC}}$, …
  - simulation
  - P2P-based authorization recycling
- future work
  - publish-subscribe for authorizations
  - speculative authorizations