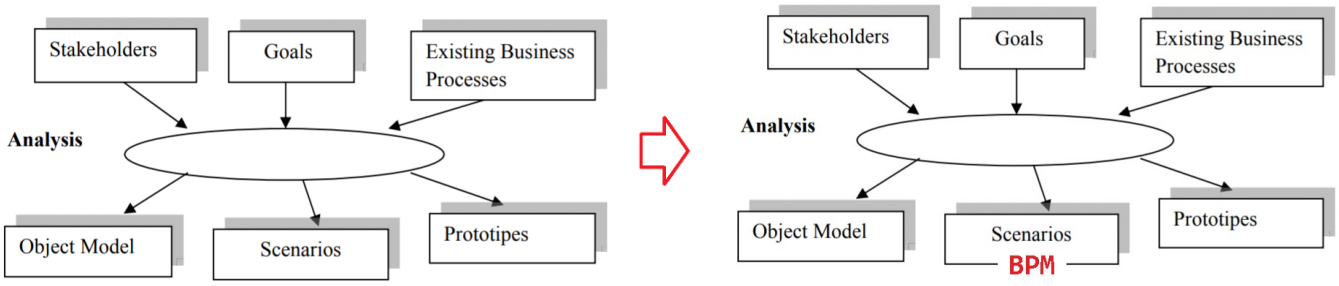
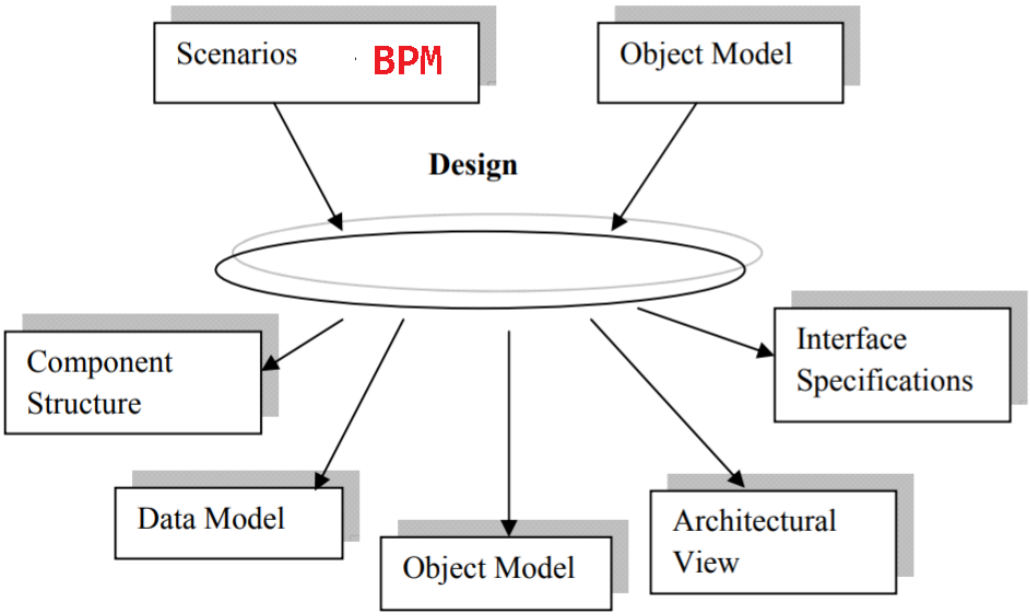
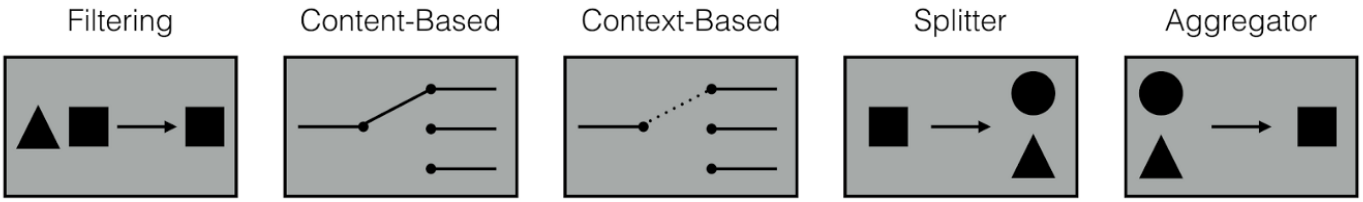
```
const { fromEvent } = Rx.Observable;
const target = document.querySelector('.box');

const mouseup = fromEvent(target, 'mouseup');
const mousemove = fromEvent(document, 'mousemove');
const mousedown = fromEvent(target, 'mousedown');

const mousedrag = mousedown.selectMany((md) => {

  return mousemove.map((mm) => {

    return {...};
  }).takeUntil(mouseup);
});

const subscription = mousedrag.subscribe((pos) => {
  target.style.top = pos.top + 'px';
  target.style.left = pos.left + 'px';
});
```

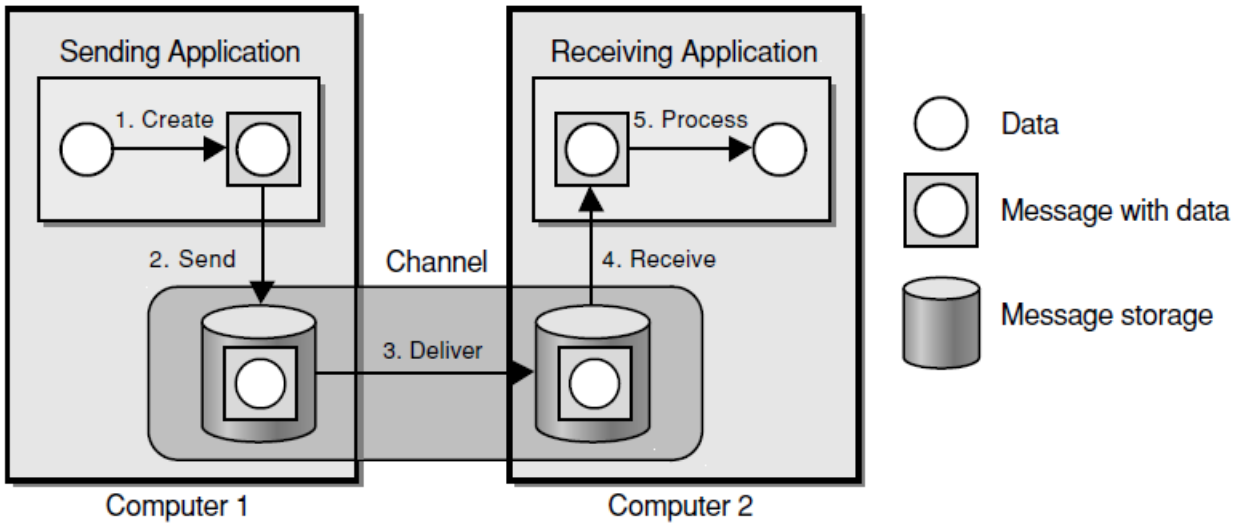
```
const { fromEvent } = Rx.Observable;
const target = document.querySelector('.box');

const mouseup = fromEvent(target, 'mouseup');
const mousemove = fromEvent(document, 'mousemove');
const mousedown = fromEvent(target, 'mousedown');

const mousedrag = mousedown.selectMany((md) => {
  const startX = md.clientX + window.scrollX,
        startY = md.clientY + window.scrollY,
        startLeft = parseInt(md.target.style.left, 10) || 0,
        startTop = parseInt(md.target.style.top, 10) || 0;

  return mousemove.map((mm) => {
    mm.preventDefault();

    return {
      left: startLeft + mm.clientX - startX,
      top: startTop + mm.clientY - startY
    };
  }).takeUntil(mouseup);
});

const subscription = mousedrag.subscribe((pos) => {
  target.style.top = pos.top + 'px';
  target.style.left = pos.left + 'px';
});
```



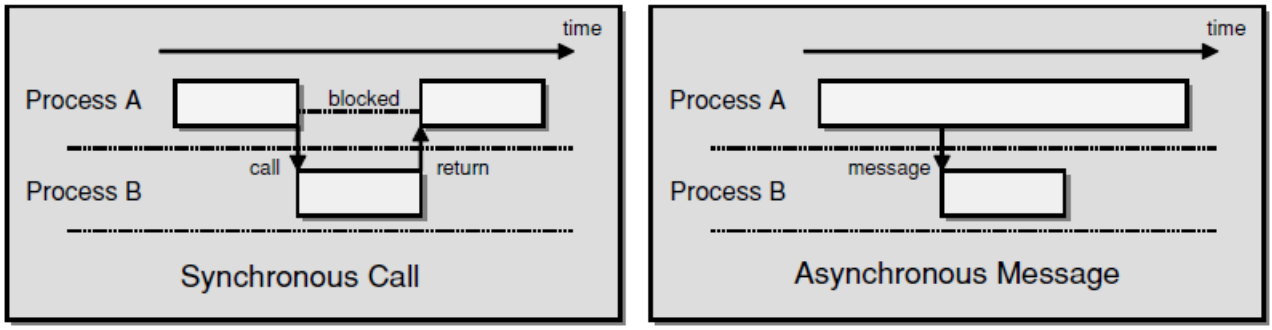


Iterace vývoje produktu

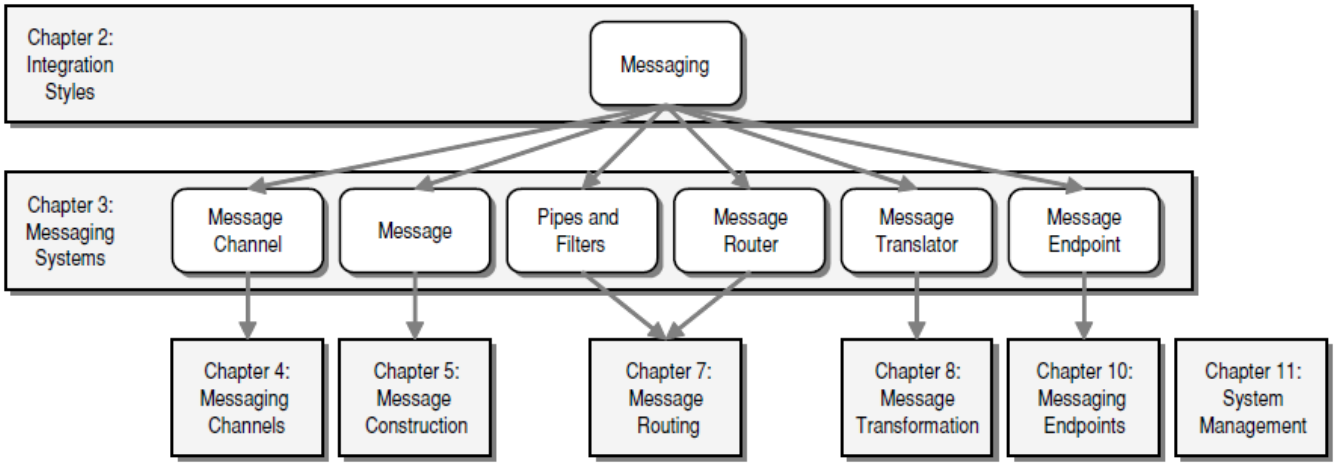




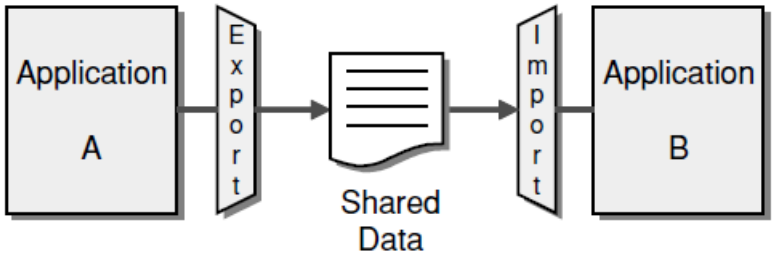
Message Transmission Step-by-Step

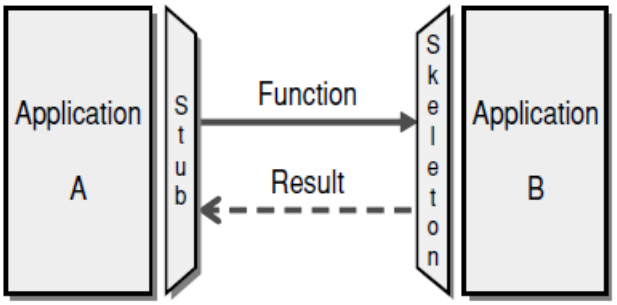


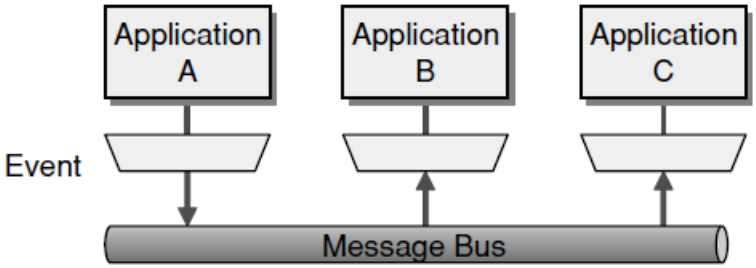
Synchronous and Asynchronous Call Semantics

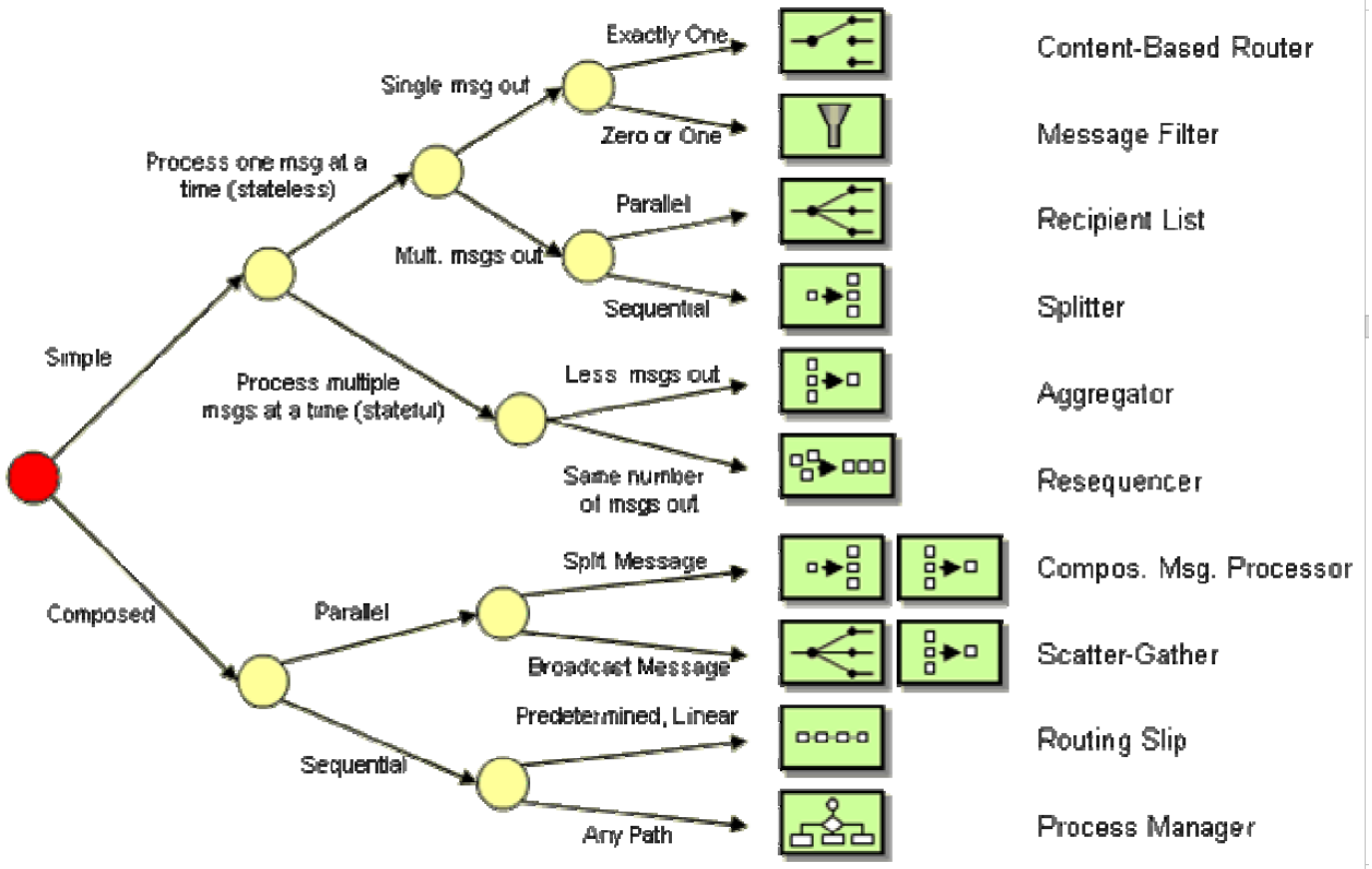


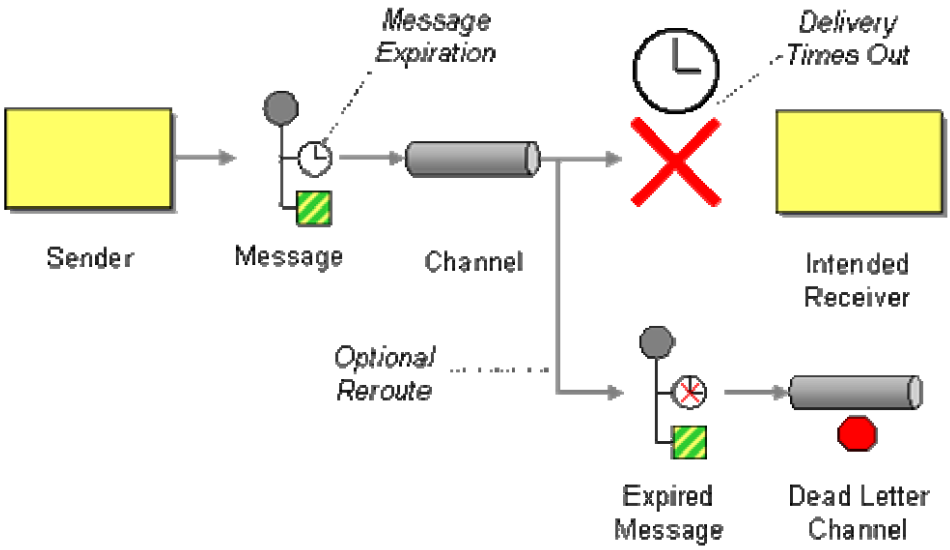
Relationship of Root Patterns and Chapters

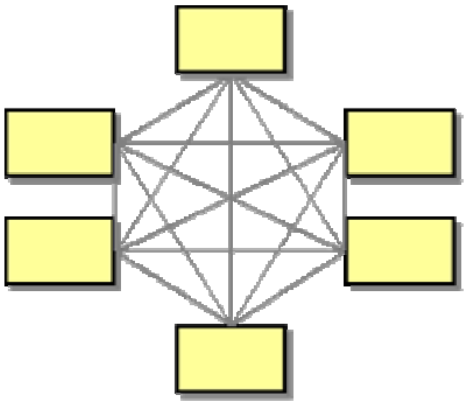




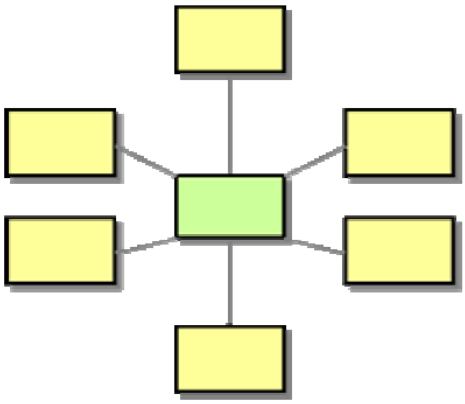



Simple
Process one msg at a time (stateless)
Single msg out
Exactly One
Zero or One
Mult. msgs out
Parallel
Sequential
Process multiple msgs at a tune (statetul)
Less msgs out
Same number of msgs out
Composed
Parallel
Split Message
Broadcast Message
Sequential
Predetermined, Linear
Any Path
Content-Based Router
Message Filter
Recipient List
Splitter
Aggregator
Resequencer
Compos. Msg. Processor
Scatter-Gather
Routing Slip
Process Manager

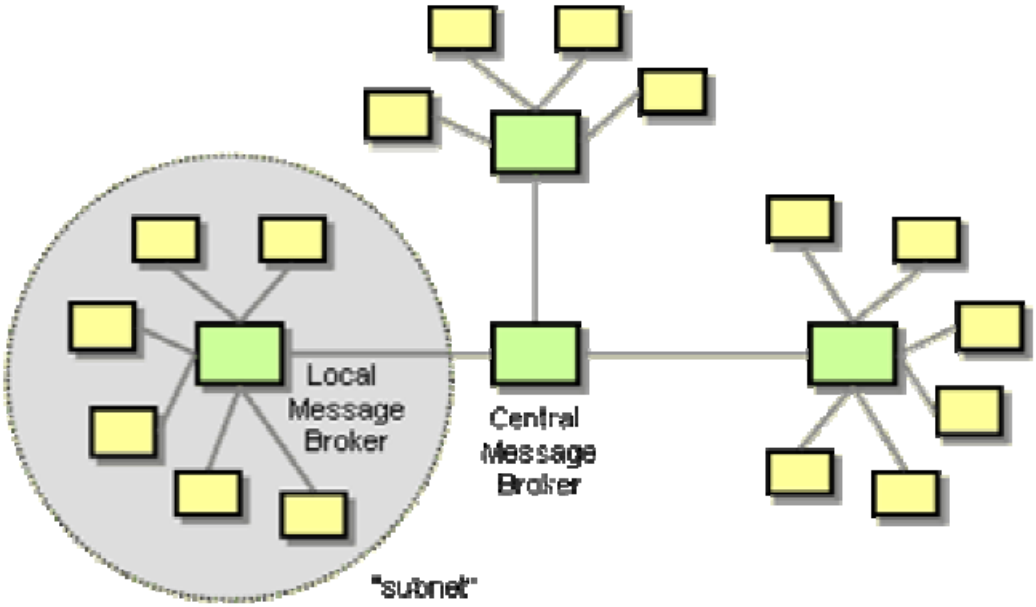

Message Expiration
Delivery Times Out
Sender
Message
Channel
Intended Receiver
Optional Reroute
Expired Message
Dead Letter Channel



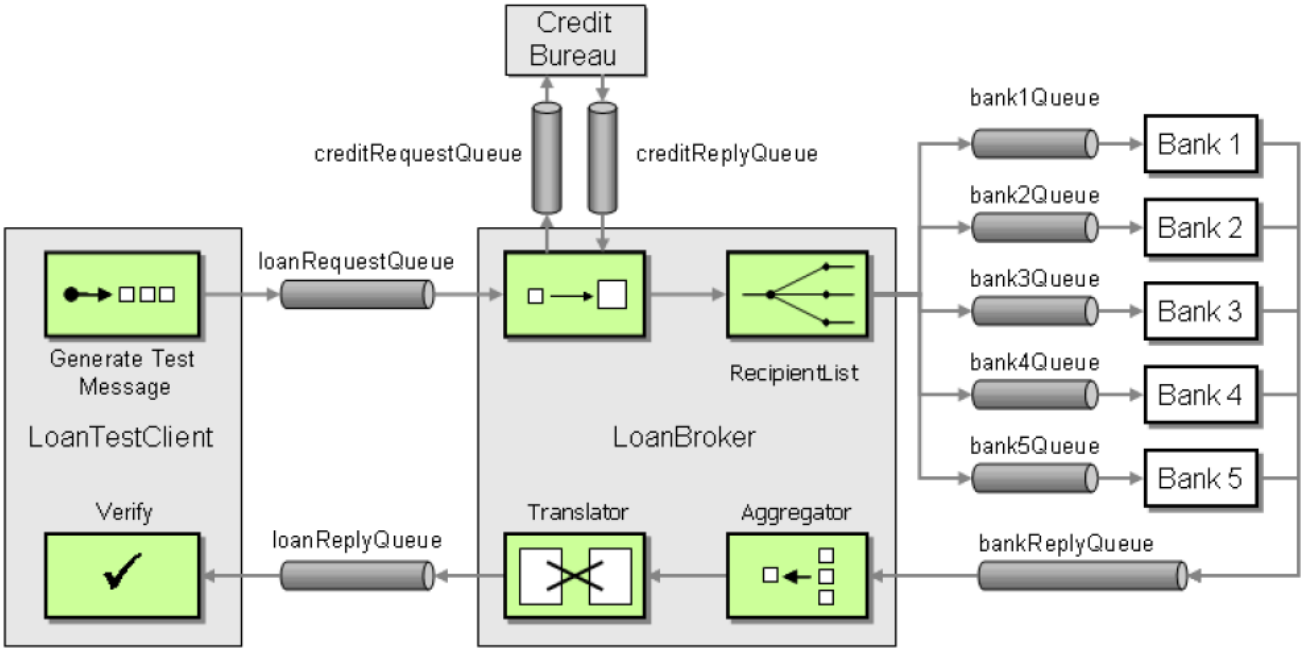
Integration Spaghetti as a Result of Point-to-Point Connections



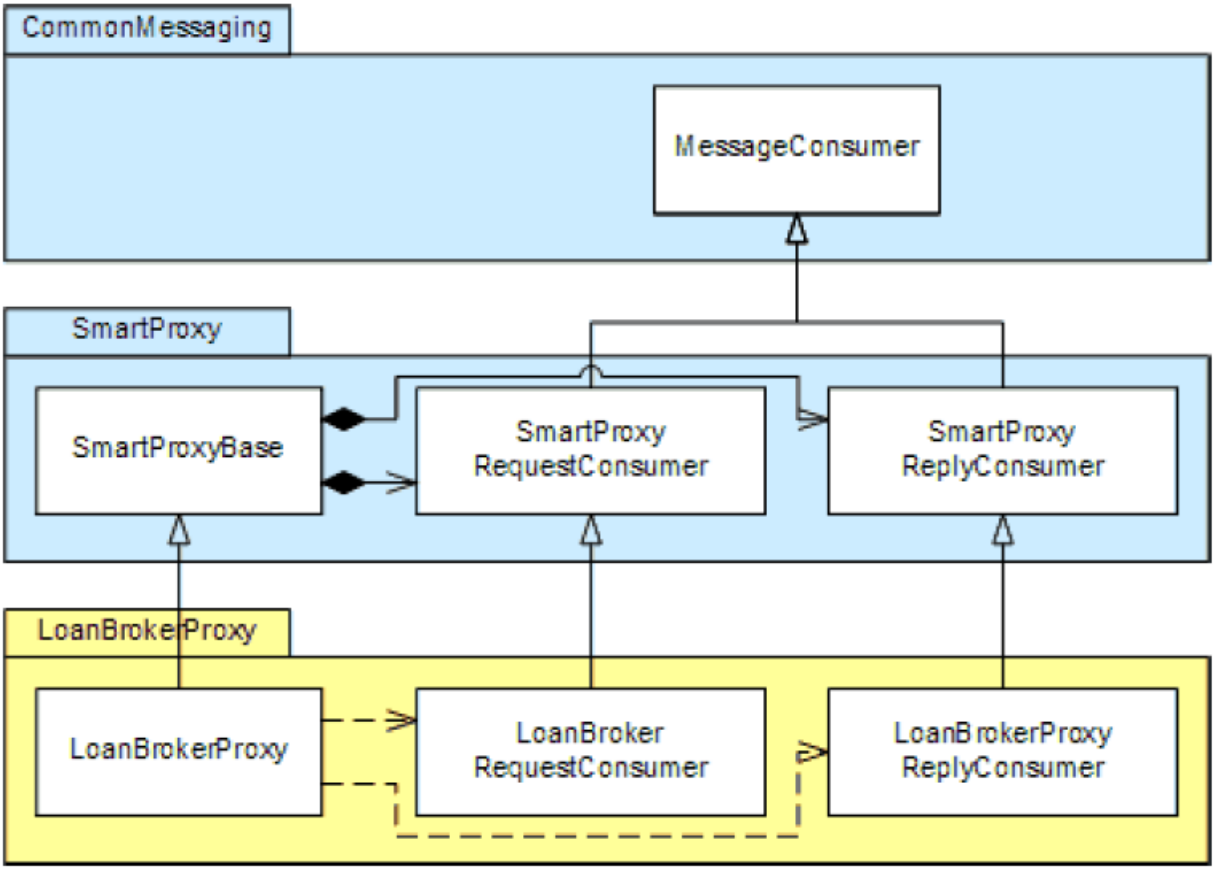
Using a central *Message Broker* :



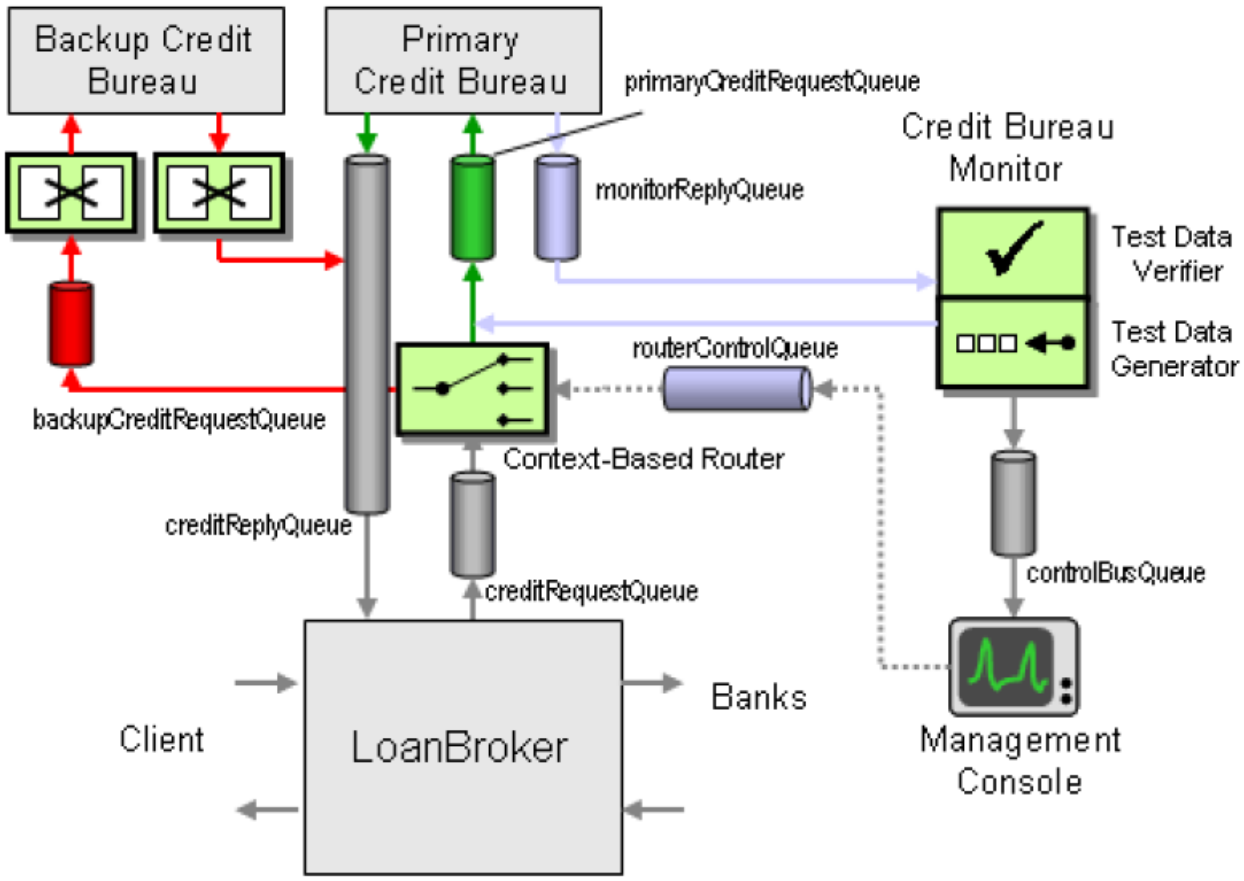
A Hierarchy of Message Brokers Provides Decoupling while Avoiding the "Über-Broker"



Loan Broker with Message Queue Interfaces



Loan Broker Smart Proxy Class Diagram



Explicit Failover with a Context-Based Router