

# Equity – transparent and live risk assessment

Tackling complexity in the heart of Finance  
QCon London 2013, #1685, Tormod Varhaugvik



Tax Norway



# The Governmental Financial Institution

- Tax Norway
  - 110 billion € in revenue from a population of 5 million
  - 610 million € in operating cost
  - 6.000 employees, 10.000 users, supported by 120 systems
  - 700 working in IT
- Status
  - Enterprise Architecture program started 2009
  - This Architecture was defined and committed in 2010
  - We have managed to change a large Government Organisation
  - Major projects are now building the future
  - We are establishing a private Cloud
- Me
  - Some sort of Architect in the Enterprise Architecture practice
  - Technical background, Enterprise Application domain since 1993



- Your Challenge
- Tax Norway' Challenge
- Strategic Approach
- Software Design
- Architecture Blueprint



# Your Challenge

*“...policymakers must insist that the large financial firms that they supervise, be capable of monitoring and managing their risks in a timely manner and on an enterprise-wide basis.”*

—Ben S. Bernanke, Federal Reserve Chairman  
March 2009



# Tax Norway' Challenge

Why do you complain? we are totally regulated...



# Our Tax System

- 5 lines of business;
  - Personal Tax
  - Company Tax
  - VAT
  - Inheritance Tax
  - Employer Tax

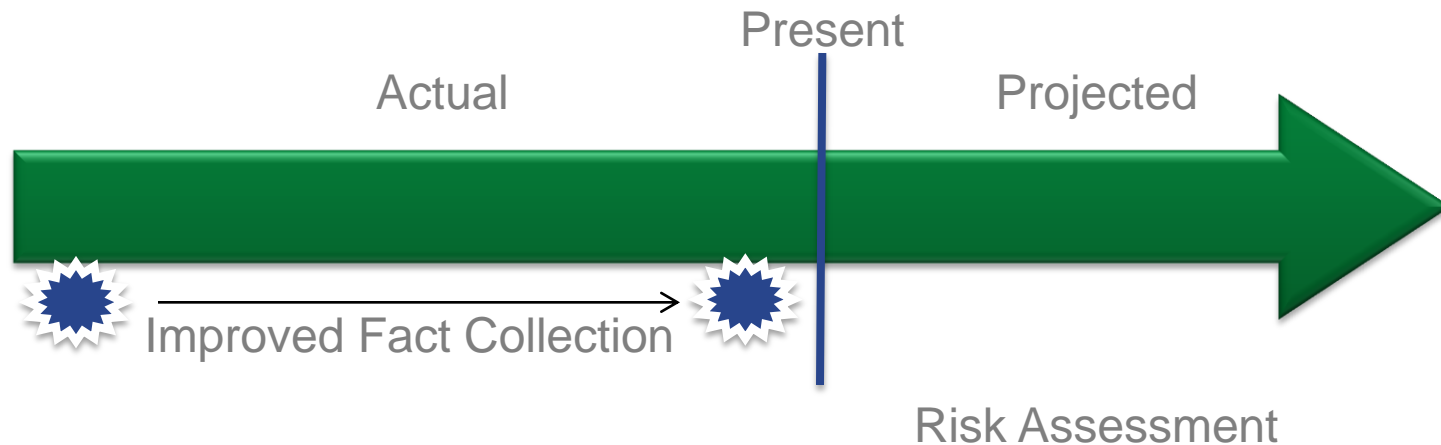
A Tax Form is a  
Financial Product

We calculate  
Taxable Amount,  
you calculate Equity

- Personal Tax is a complex domain:
  - Asset-info directly from 300.000 reporters: Salary, Taxable goods, Savings and interest, Stocks and bonds, Gifts, Insurance, Day care of children (8.000), Vehicles, Real estate, Charity, ...
  - Approx. 50 sub-forms, 5.000 unique data types and 7.000 rules
  - Ends up in 8-14 different taxes collected together
  - Approx. 75% accepts tax without participation. Deliver by SMS
- Taxable income, tax, and fortune is public information
- Accurate calculation: The public trust us in doing it right



# Actual vs. Projected



- The Future is still unknown
- Improve by having timely facts
- A timeline of Party behaviour
- Correlations between different Lines of Business
- High flexibility in new models for Fraud Detection
- Fraud Detection is really small scale Risk Assessment



# The Kinder Surprise?

- All lines of business
- All specific yearly Rules and Information
- Agile to New Business Capacities
- Audit and Proof of Procedure
- Much more Data
- Up-to-date and Event Driven
- 24/7, Straight Through and Workflow
- Easier to Maintain and Understand
- Excellent Sourcing
- Much Lower Cost of Ownership
- Much Faster
- Migration Friendly



This would break  
in any  
classic approach



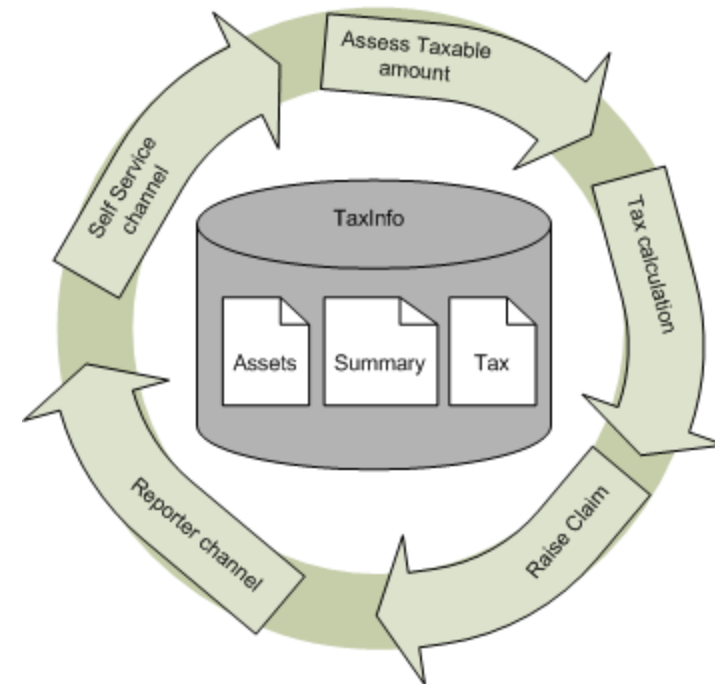


# Strategic approach



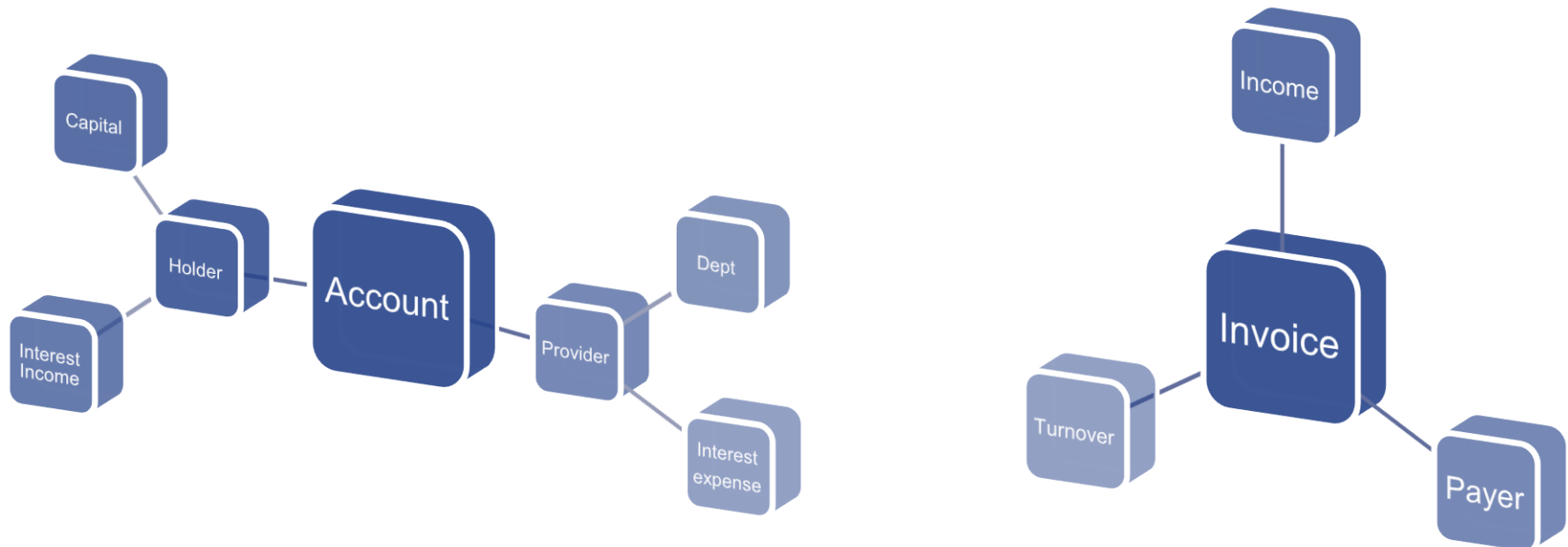
# Continual Aggregate Hub

- Big Data repository of Documents
  - Loosely coupled
  - Immutable & versioned (legislation)
  - All lines of business side-by-side
  - Yearly legislation side-by-side
  - Simple for 24/7 usage
  - Search engine
  - Access control
- 
- In-memory processing layer
  - A Module consists of business logic and its GUI
  - A Module own the document type it produce
  - Each line of business have a dedicated set of Modules
  - Straight through processing and workflow in one
  - There are many running versions of Modules and Processes





# Asset Dimensions

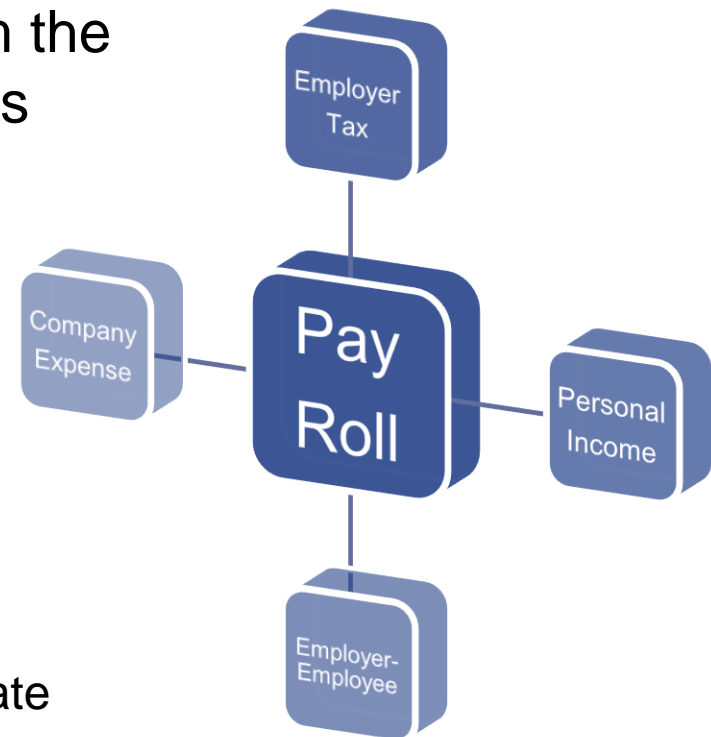


- Asset – multi-dimensional properties
- Trees and Chains can easily be analysed by computers
- Main organisational focus: get the Assets right



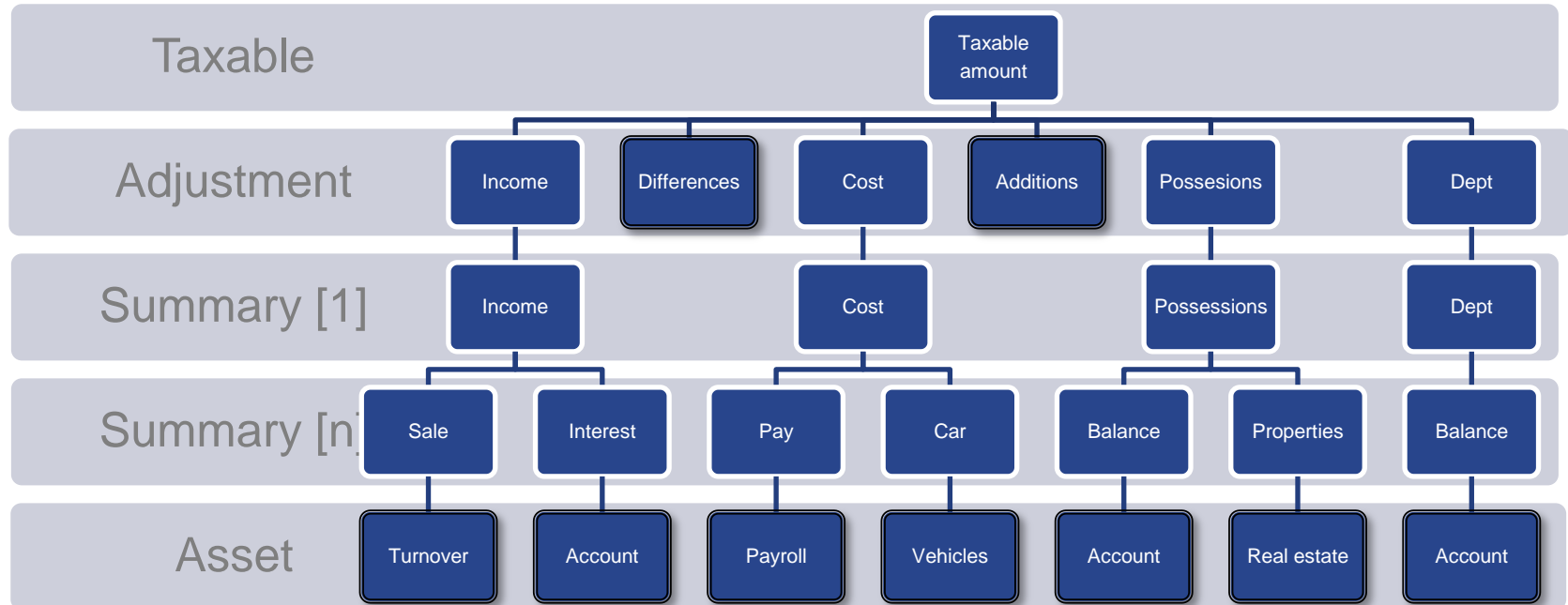
# Live Payroll initiative

- Salary is reported directly to us from the payroll systems, as they pay salaries
- We calculate Employer Tax
- We then also know:
  - Companies salary expenses
  - Persons income (solvency)
  - Employment rate (almost)
  - Benefit calculation use salary fact
  - Public Sector re-use
  - Employee knows if Employer is legitimate
- 120.000.000 documents pr. year
- Everybody applauds this





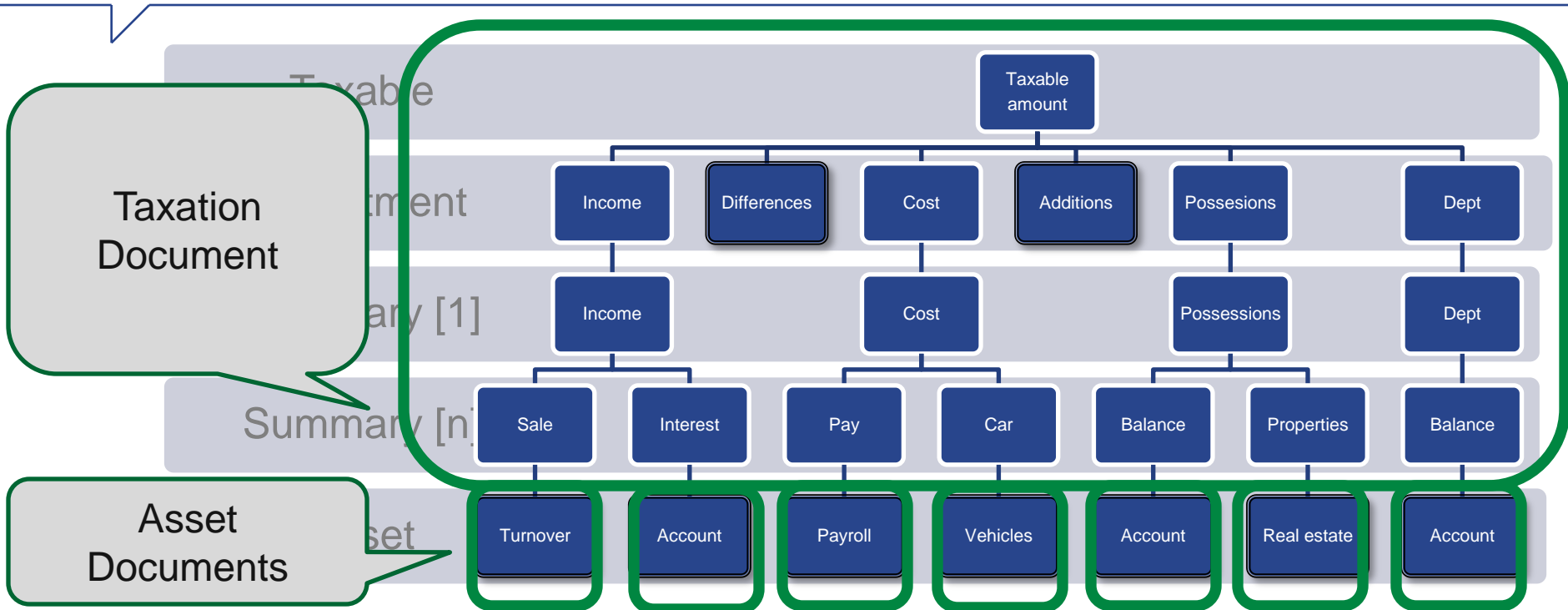
# Assets and Summaries



- A Summary (form / agreement) pr. Party. (Party is where the risk is)
- An Asset may be included in many Summaries
- High flexibility on making new Summaries (tactical)
- Summaries without Assets are less credible, and represents an Audit challenge



# Assets and Summaries



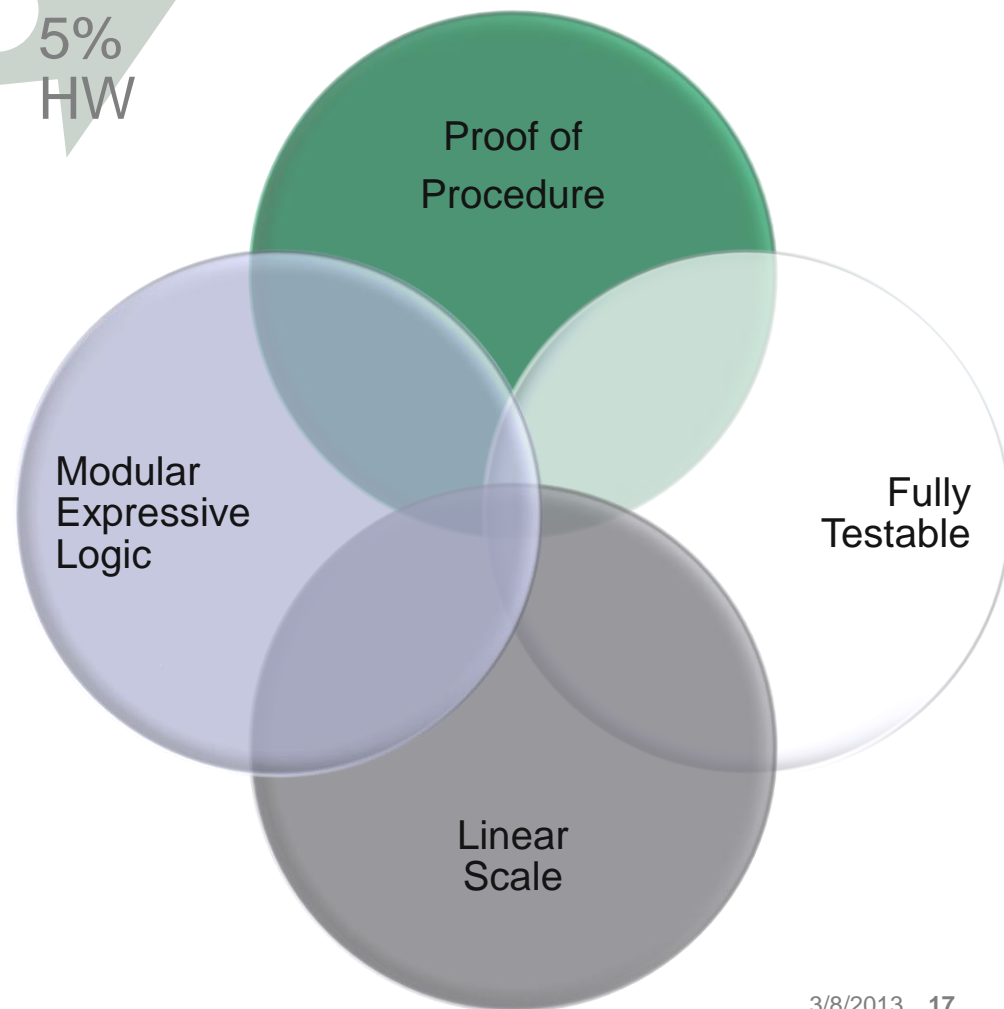
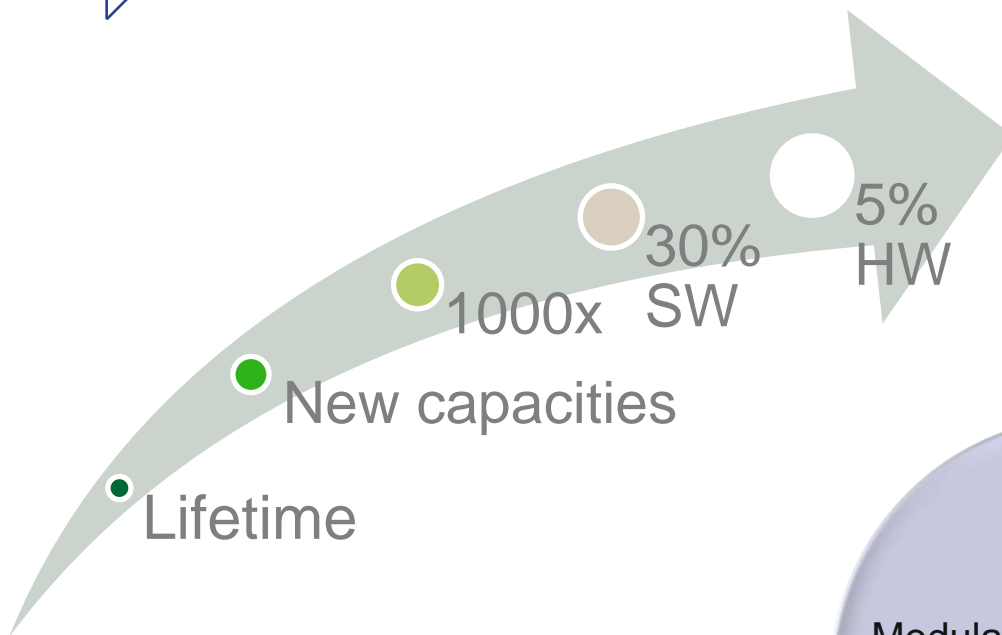
- Assets and Summaries are treated as independent documents
- The documents are versioned and has a comparable timeline
- The documents can be electronically signed
- The documents are legislated Proof of Procedure
- The documents are a Business Event and its Contents



# Software Design



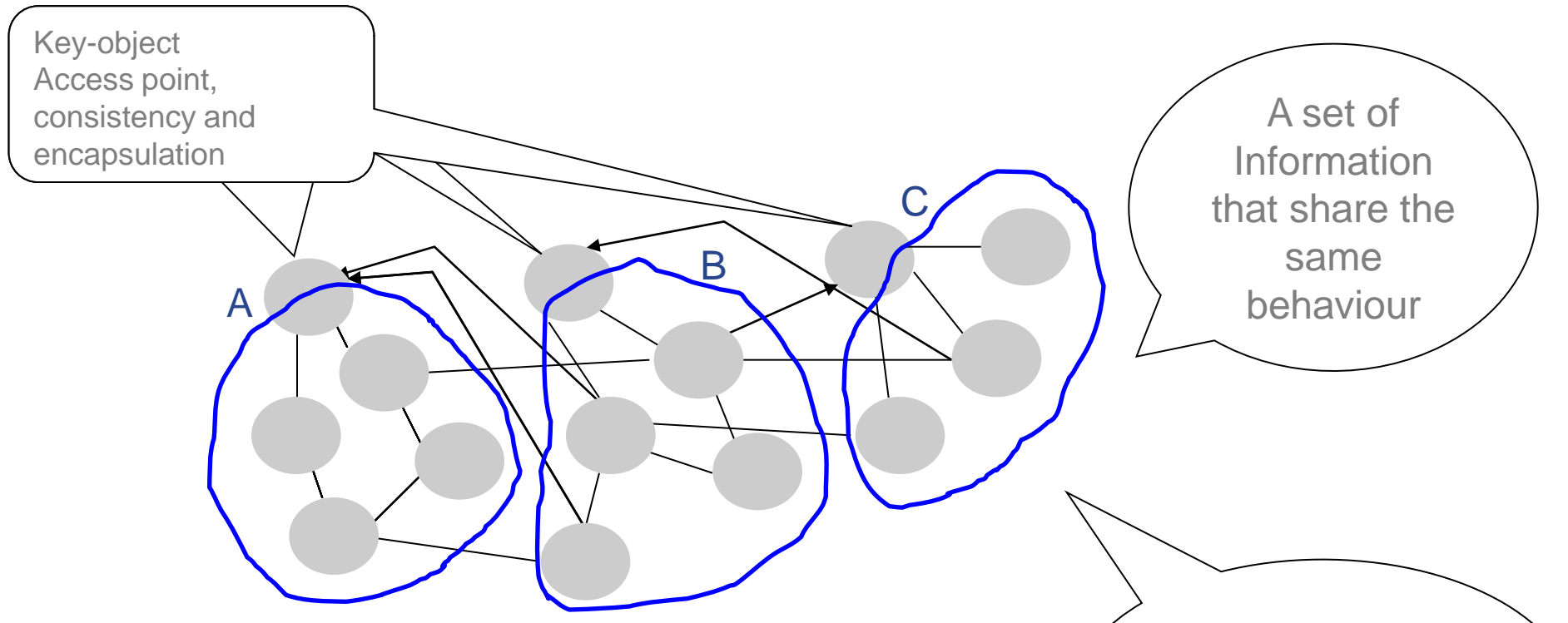
# Design Goals







# Decomposition – "Aggregate design"



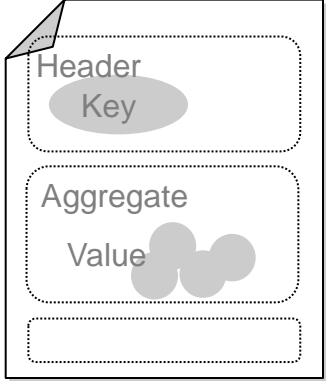
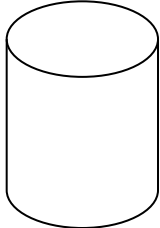
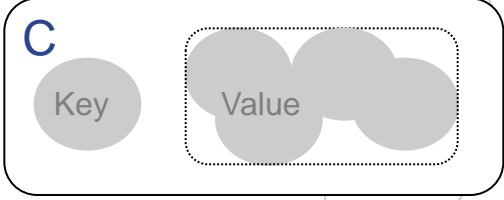
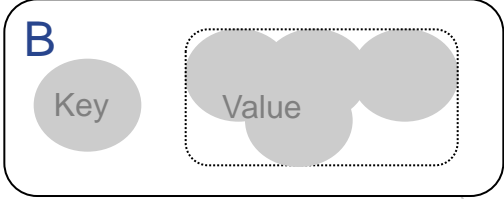
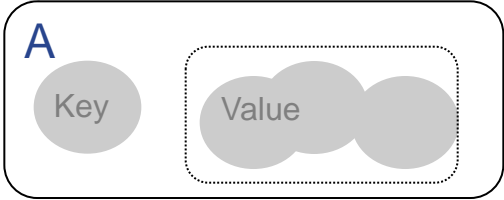
- This is really just good software design
- Good service orientation
- Maintainable and testable components
- Independent information sets
- Independence premise for parallelism



# The In-memory and Big Data fit

Memory and processing  
comprise many machines  
Disk storage detached  
from the application

Application

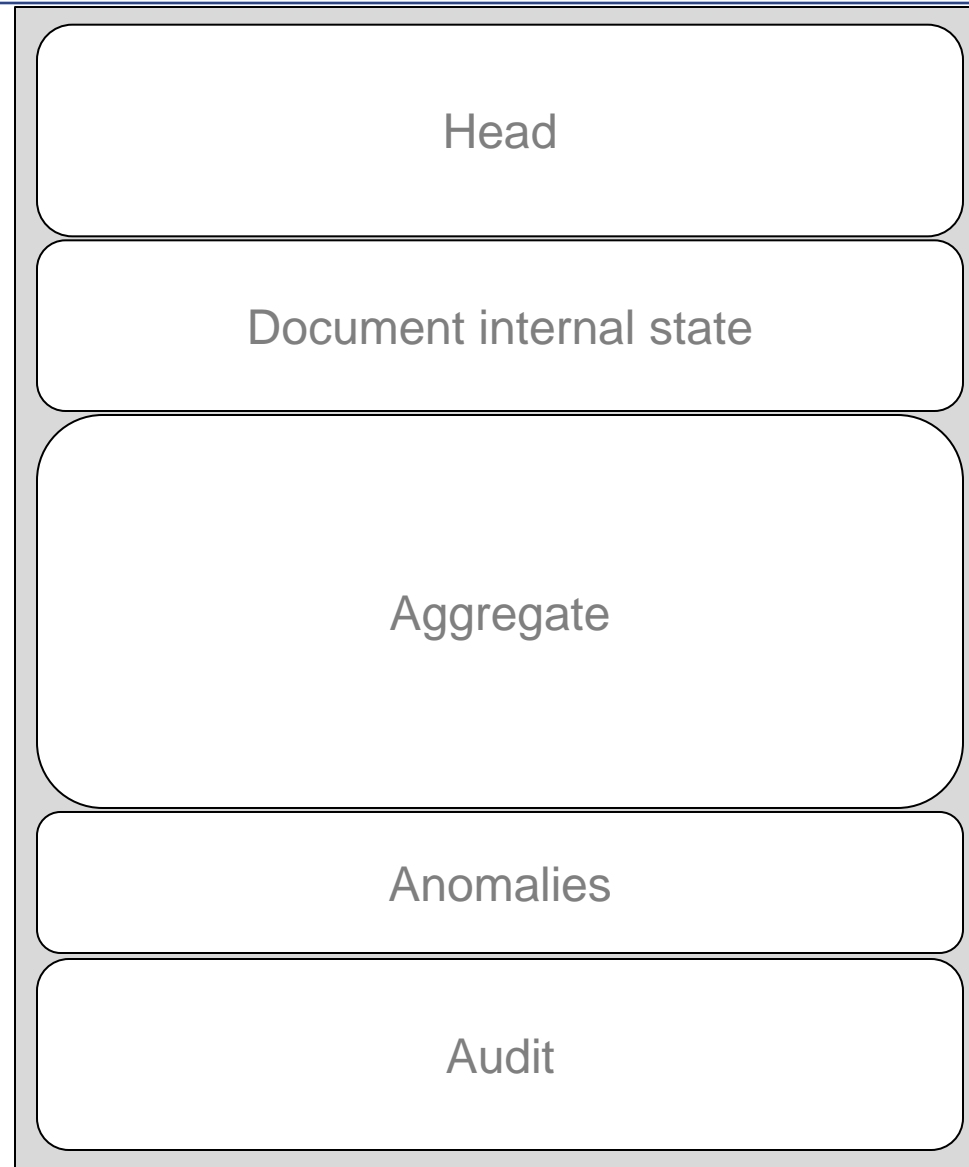


- Application
  - Information composition
  - Business logic
- Key object may be complex
- Application is free from deployment concerns
- Big Data



# The XML-document – Master template

- The Head is
  - key object
  - classification of information
  - also a protocol and interface
- The Head is to the Repository as a Library Catalogue Card is to a Library
- Robust and Consistent
- Independent and Shardable
- Reduced I/O and Concurrent
- Historically Correct
- Business Event and Data
- External XML-schemas
- Search Engine
- (Only one producer, this is no database system!)





# The XML-document – Tax Assessment form

- Main subjects are debit and credit of a transaction, under what legislation, at what time, and who we should trust this information
- Immutable when 'public'
- New version on update
- Transparency by referencing underlying documents
- Consistency by referencing underlying documents
- Complete audit in the same document
- Insight without business logic
- The interface to any consumer

GUID, timestamp,  
 concerns, state [private, public, deleted, replaced]  
 reported by, replaced by → GUID  
 datatype, legitimate period [income year, date period],

phase [prognosis, prefilled, delivered, assessed, complaint]  
 version  
 module state [new, manual handling, finished]

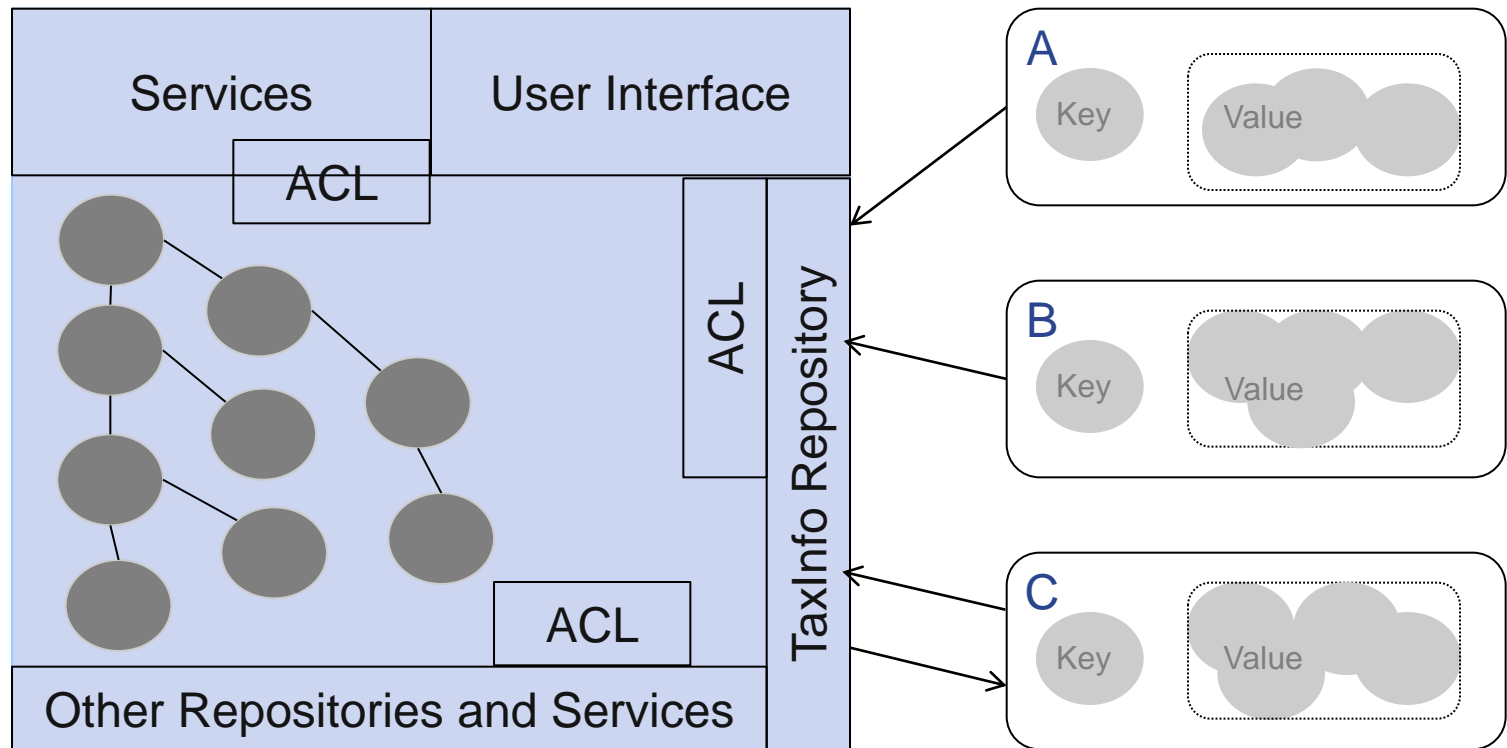
field2.1.1 text  
 value  
 ref GUID → GUID  
 field3.1.12.7  
 ...  
 field5

description  
 concerns fields → this.URI

user name  
 timestamp  
 event, reason  
 concerns fields → this.URI



# The Application – Cloud Enabled

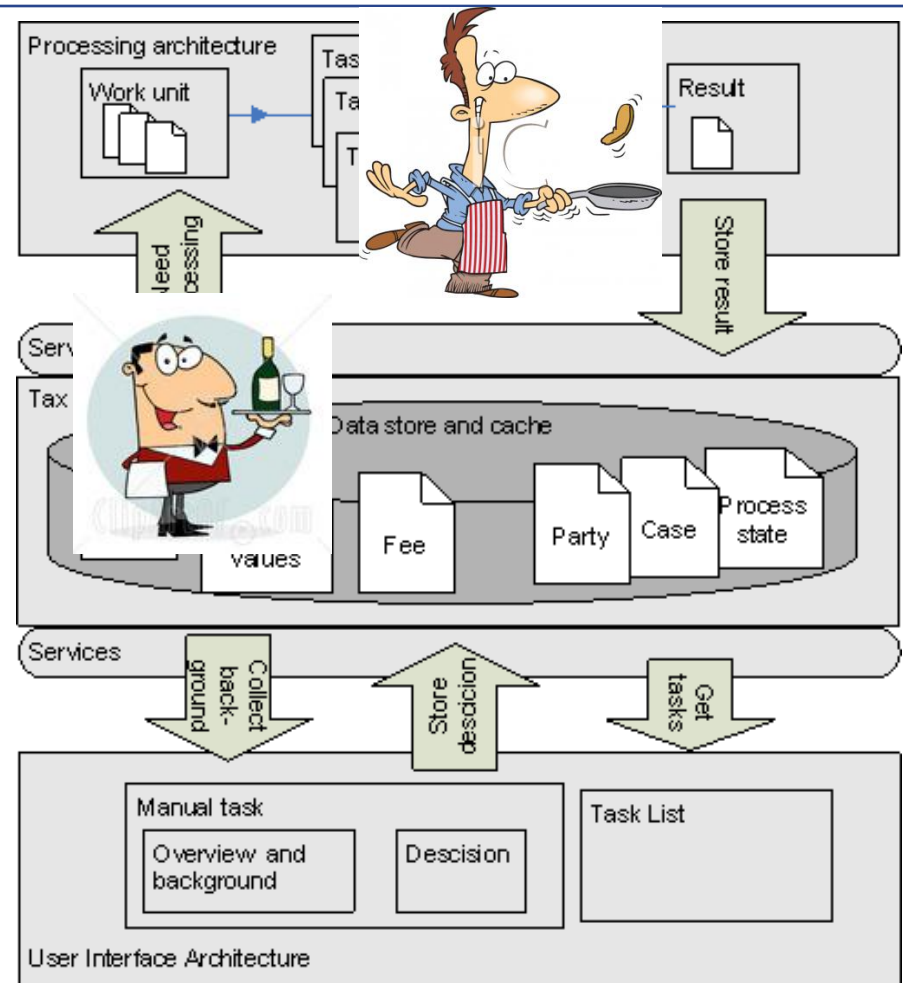


- The Business Logic run here
- Overly simplified, but still illustrates that information is taken out of their coarse documents, and - through an Anti Corruption Layer -, structured in a specific Domain
- Eventually Consistent: Comparing last version of C to new version of C as a consequence of changes to A or B is vital



# Straight through processing and Workflow

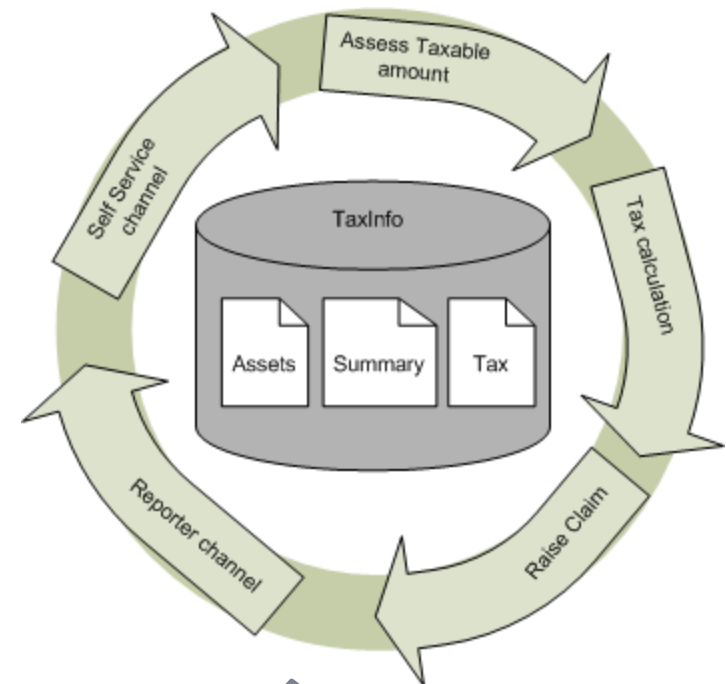
- **Component based / layered**
  - Separate business logic from architecture
  - Re-use of services, GUI and data
- **Process focus**
  - Straight through and Workflow
  - Parallel
  - Linear scale
- **Open standards**
  - xml, java, container, web
  - Sourcing
  - Lifetime principle
- **Object oriented**
  - Explicit Domain Model
  - Rich semantics, DSL
  - xml 1:1 with java
- **Test and simulate**
  - Automated tests
  - White-box test
  - Dry run
- **Operations**
  - Re-run after fault





# The Continual Aggregate Hub

- Design patterns:
  - Domain Driven Design
  - Tuple Space, CQRS, BASE, SOA, ODS
  - XML-documents, plain Java and REST
- Migration path by having old (deprecated) and new participate.
- Consume and produce at systems pace
- Continual process, raise claim at designated time or event
- A timeline for every document
- Short time-to-marked
- Refactor at any time
- Search and view of all documents
- 360° of Parties Assets, Forms ++



100' of document types  
100' of applications  
Cloud enabled  
Vast deployment options



# Blueprint





# A Blueprint of your new Core System?

- Structurally comparable, even though data and logic are different
  - Handling of Financial Documents in one architecture (Securities, Assets, ...)
  - Consolidate Parties
  - Have all Business areas publish to “Continual Aggregate Hub”
  - Tap into “feeds”
  - Build live Risk Assessment
  - Modernise systems prioritised by Business need
    - Modules supporting Business
    - Documents contains Products
  - Yes, you will have to Run the Gauntlet (each line of Business, IT dept., CTO, and CIO)
- PoC results
  - 80.000.000 Documents
  - 5.000.000 Party Summaries
  - 8 Risk dimensions
  - 3 minutes on 12 servers
  - € 50.000 in Hardware
  - € 130.000 in Licence
  - 30% of current Maintenance
  - 5% of current Hardware
  - Best Sourcing there is



# Thank you!

- [http://domaindrivendesign.org/library/vernon\\_2011](http://domaindrivendesign.org/library/vernon_2011)
- <http://www.infoq.com/minibooks/domain-driven-design-quickly>
- <http://www.bankofengland.co.uk/publications/Pages/default.aspx>
- <http://www.riskresearch.org/>
- <http://tormodv.blogspot.com/2011/02/comment-on-restful-soa-or-domain-driven.html>
- <http://tormodv.blogspot.com/2010/11/concept-for-datastore-and-processing.html>
- <http://tormodv.blogspot.com/2011/02/document-store-for-enterprise.html>
- <http://tormodv.blogspot.com/2012/01/tax-norways-poc-results.html>
- <http://tormodv.blogspot.com/2011/09/dont-let-enterprise-service-bus-lead-to.html>
- <http://tormodv.blogspot.com/2013/01/target-architecture-looking-good.html>
- <http://www.slideshare.net/tormodv>

*My blogs are written for stakeholders and architects, and meant to be as timeless as possible.*



# Background



# 8-machine strategy

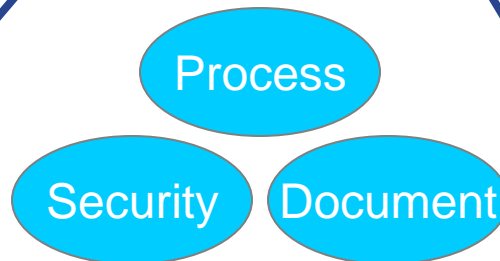
External communication

Party

Compliance

A landscape of federated event-driven systems

Build or buy?  
Clear separation.  
Migration friendly.



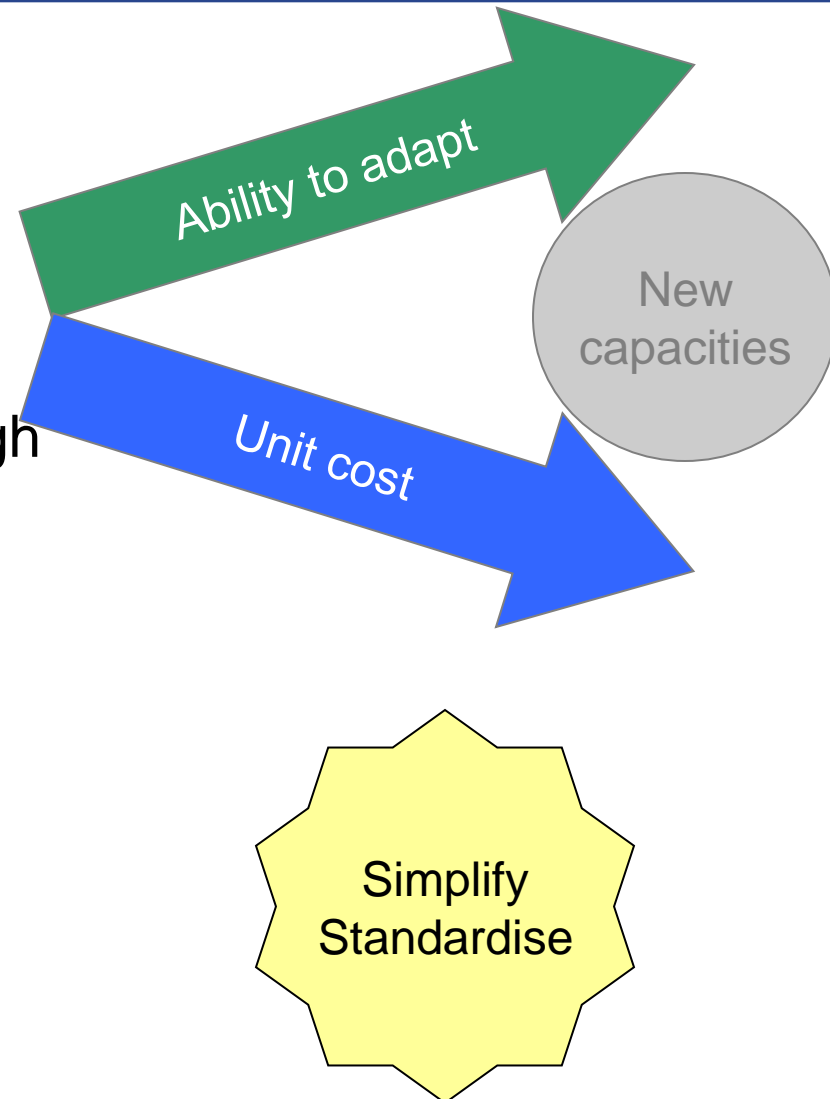
Assessment

Collection



# Our challenge

- Party centric
- Self service
- Re-use
- Integrity, Transparency
- 24/7, Event driven, Straight through
  
- Yearly revised legislation
- Long lived (10+ years)
- Business challenge: Live payroll
  
- Sourcing and Migration
- In-memory, Big Data
- IaaS, PaaS, SaaS





# Classic Situation at “any” Organisation

- Systems pr. line of business
- Systems pr. subsidiary
- Who are the Parties?
- What are the Assets?
  
- These large systems have too many concerns
- Self centric, untestable
- Not able to bear consistency, uptime, performance and ability to adapt
- Size if not a problem by itself, but the objective is
  
- Stop feeding these systems!
- You can't wrap them in a tracksuit and expect an athlete!



(C) Monthly Pyton