

PillarOne – Risk Management meets Open Source



SAA Meeting August 29, 2008

Markus Stricker, Stefan Zumsteg, Intuitive Collaboration



- PillarOne
 - Scope
 - Why Open Source?
 - What is commercial Open Source?
- Reserving
- Modelling

- PillarOne is
 - a **community** to exchange views, methods and discuss issue concerning enterprise risk management
 - An enterprise **software platform** to build actuarial applications, e.g. reserving, pricing, risk management
 - driven by a **spirit of sharing** – open source

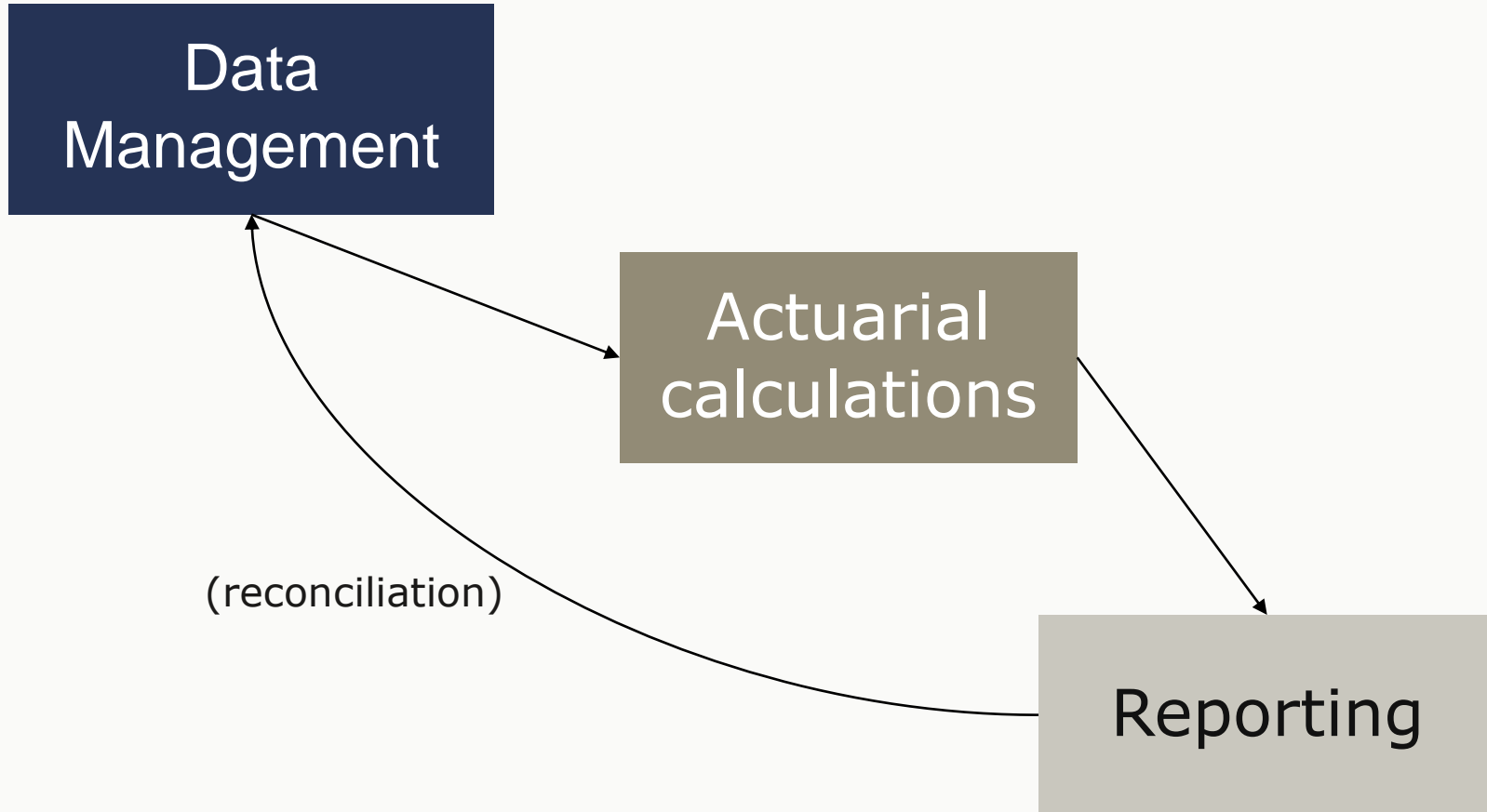
Why Open Source?

- Transparency is an asset in risk management; actuaries want and should know how things are done
- An open, collaborative process is the best way to empower users to shape products and how they are developed
- Open source allows the broadest form of collaboration; even of competitors
- The costs are shared by different groups and firms
- Encourages an ecosystem → No dependency from a single provider
- New attitude towards intellectual property: Accessing it is free, but delivering it costs (professional services)

What is Commercial Open Source?

- There exists an ecosystem of **professional service providers**, e.g.
 - Consultants for custom extensions
 - Maintenance and support
 - Training
 - Integration with existing enterprise IT systems
- The development is **professionally managed**, i.e. project manager, developers and marketers are at least 50% working on the project and get paid for it

PillarOne is commercial open source, licensed under GPL.



PillarOne uses structured data types for all reserving objects, e.g. triangles, patterns, time series, projections.

A triangle for example is more than a collection of figures in cells.

It has

- a dated reporting interval structure (accident, underwriting year)
- a development interval structure
- a currency (where applicable)
- comments referring to the entire triangle, single cells, rows, or diagonals

→ Structured data types provide additional safety.

→ Structured data types give additional power.

Structured data types prevent

- adding monetary amounts to numbers of claims
- adding claims amount from an underwriting year triangle to those of an accident year triangle
- applying a method requiring a tail pattern to a line where none is defined
- discounting amounts in EUR with a yield curve for USD

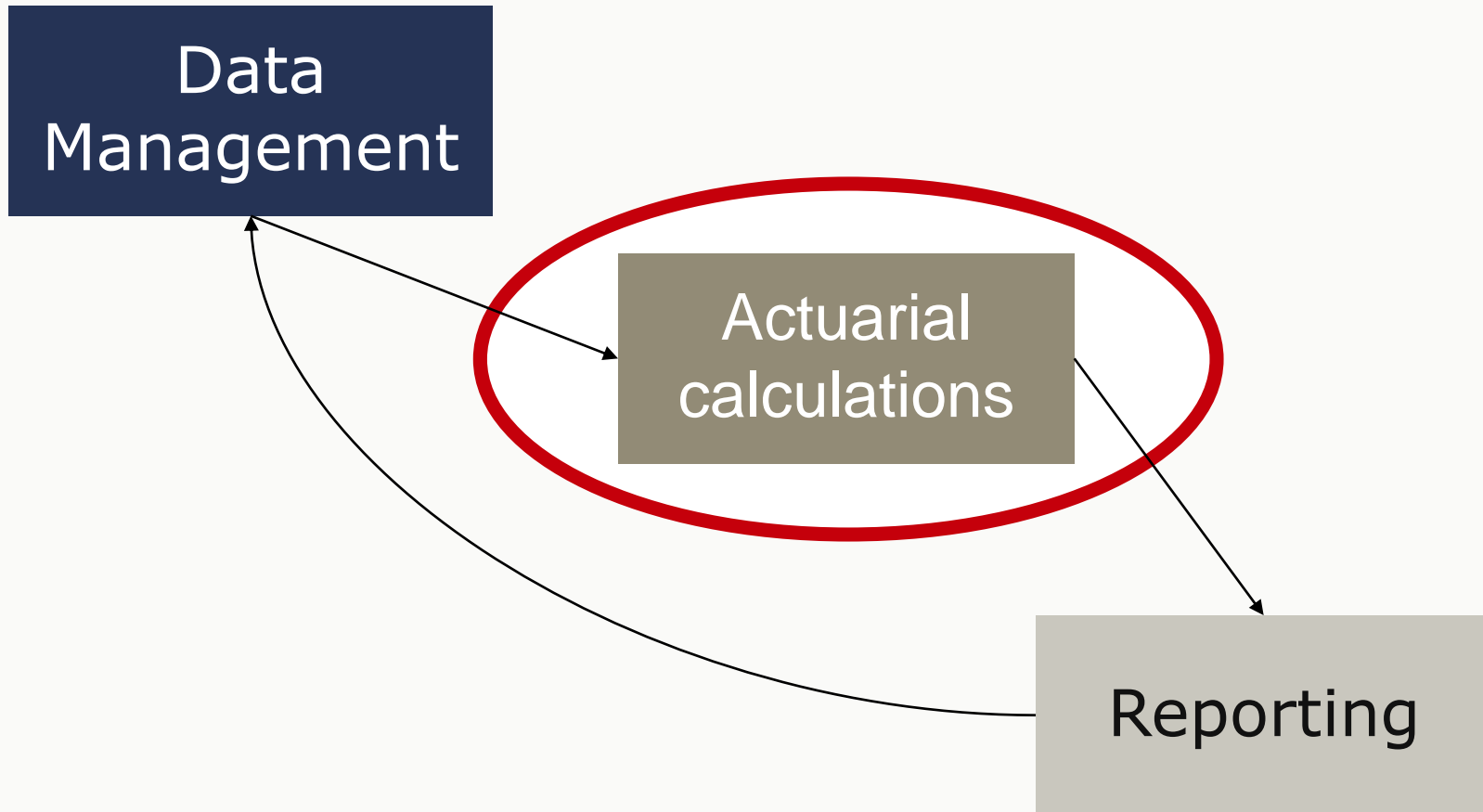
Structured data types ensure

- correct aggregation of triangles starting at different times
- ... even when they have different period lengths, as long as one is a multiple of the other

- Designed for enterprise use, data base oriented
- Native support of all major database formats. It is not just about data import:
 - The whole reserving project is a database
 - Results are objects added to the database

Advantages:

- Allows cooperative multi-user work on the same data

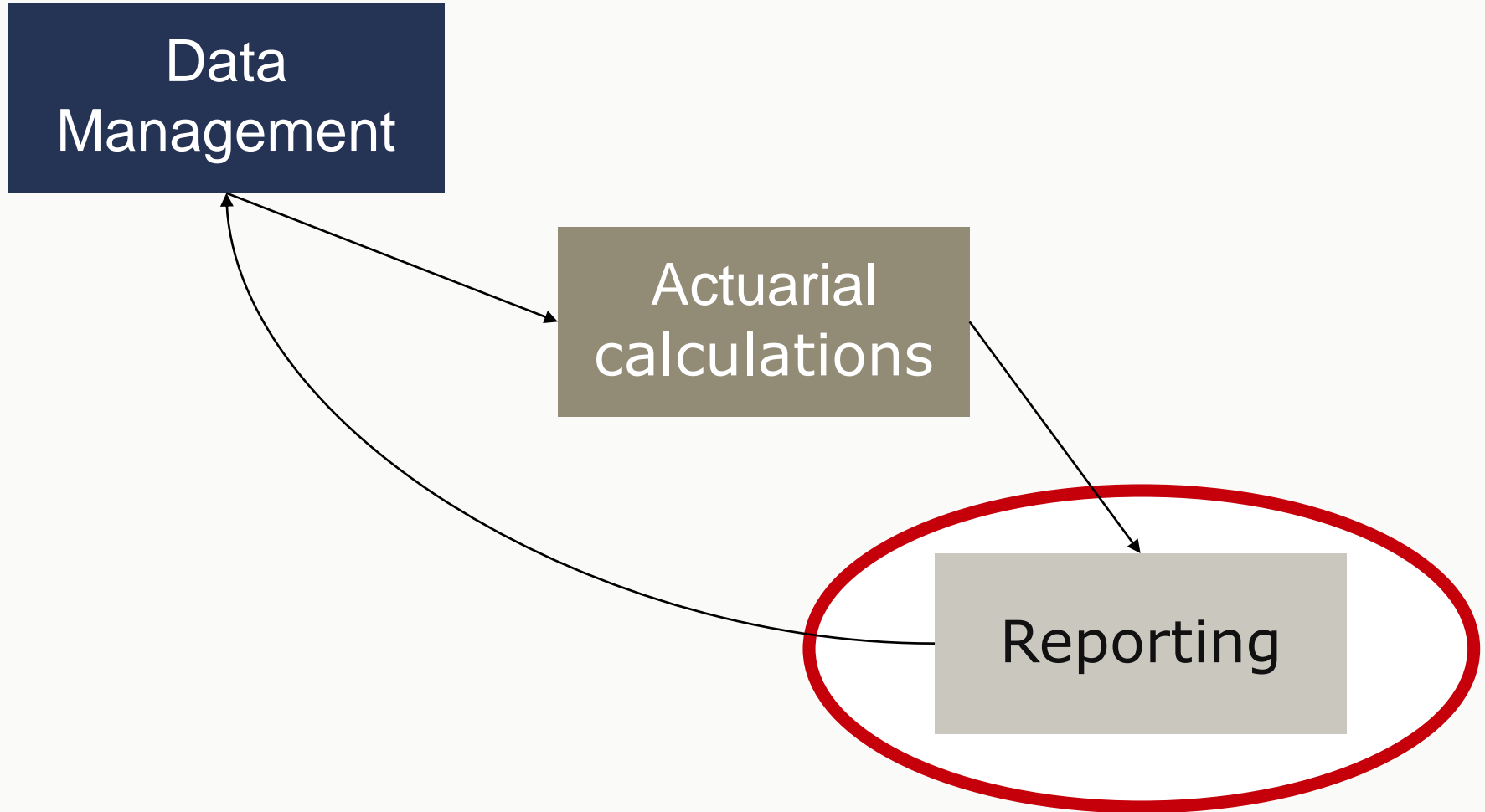


Diagnostics

- Automatic scans for common data errors e.g. negative cumulative paid
- Sensitivity analysis
- Runoff analysis

Calculation methods

- Link ratio family (Chain-Ladder, regression, etc)
- Additive model, BF
- Definition of user-defined compound methods
- Use of tail patterns
- One-click “apply to all nodes” functionality



Aggregation / Reconciliation

- Free multidimensional structuring of data, e.g. currency, continent, country, line of business, company
- Projections can be computed on aggregated node level and compared with aggregation of individual projections
- Automatic aggregation along hierarchy based on tags i.e. items with same tag are collected
- Invalidate projections if underlying data has changed (“referential integrity”)

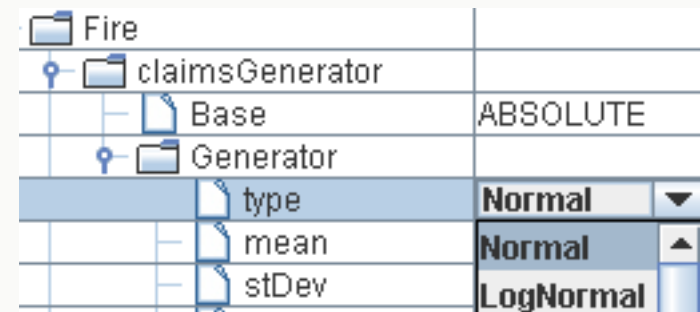
Planned improvements

- more statistical tests and analysis e.g. independence tests of reporting periods, lines of business, etc
- more pre-defined methods e.g. methods producing accident year results from underwriting year data, Munich Chain-Ladder
- easier data import from spreadsheets
- improved graphics
- “tagging” for structuring, and selective aggregation

Possible developments

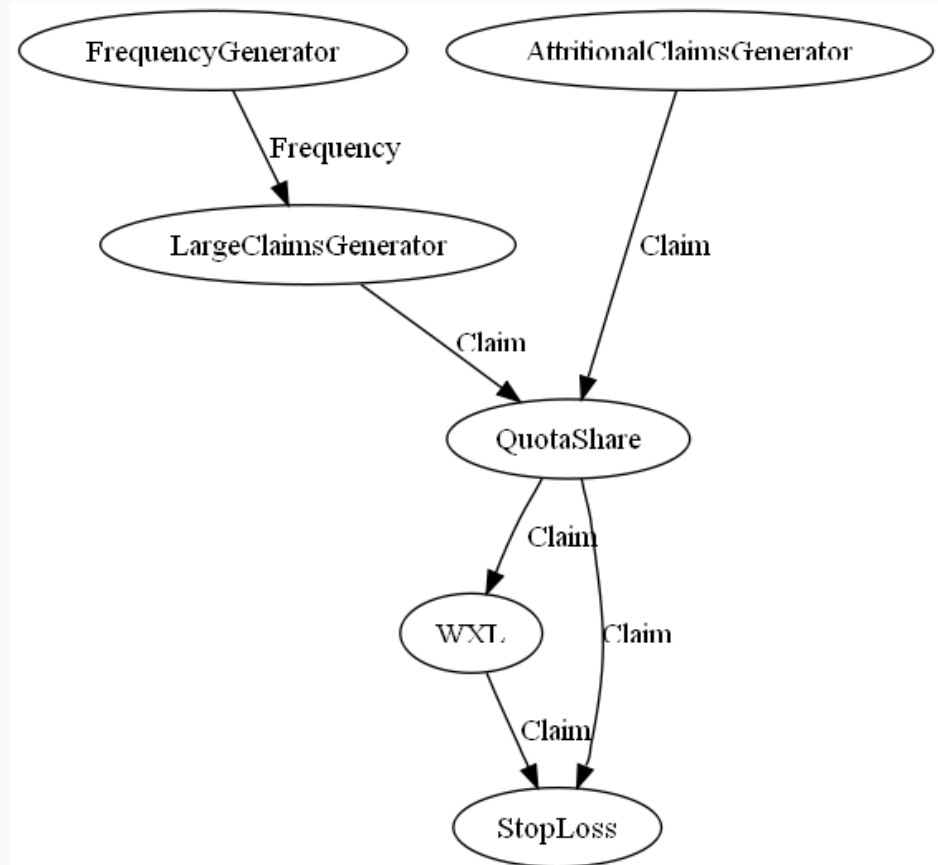
- cryptographic signatures of results and data modifications
- full audit trail
- active support of roll forward comparison of historical data, inheritance of comments
- data pre-processing steps to be combined with user defined methods, e.g. smoothing, filtering
- <Insert your suggestions here>

- Let actuaries focus on what they are best at doing: Actuarial methods and models
- Don't start each time from scratch
- The platform on which these models run is quiet helper rather than a technical jungle
 - automatically generates a GUI for a model
 - automatically generates database schema to store in/output
 - provides multi-user environment
 - makes the usage auditable
 - can scale complex models without changing any model components
 - provides a mechanism to easily add new functionality



Fire	
claimsGenerator	
Base	ABSOLUTE
Generator	
type	Normal
mean	Normal
stDev	LogNormal

- Models are “wired” components. The wires can safely transport structures, e.g. claims with all their attributes
- Rich library of standard components
- Structure information is kept separate from the model
- Input parameters are kept separate from the model
- Observers or listeners can be used to collect results, accounting, etc.



- October 2008 – Core Features:
 - build partial internal risk models
 - Wire components with a script → model
 - automatically generated GUI for parameters and results
- Q1 2009 – Reporting, Extensibility
 - slice and dice functionality (OLAP)
 - validation (parameters and models)
 - plug-in functionality

- October 2008 – P&C liability focus Solvency II context
 - standard risk capital model and application template for partial internal model in the context of Solvency II
- Q1 2009 – P&C liability focus DFA context
 - P&C liabilities: claims development, accounting

- Platform features
 - collaboration features such as comments, tags, search?
 - visual model building?
- Business logic
 - different accounting standards?
 - assets?
 - simplified: P&L, cash flow and balance sheet?
 - life liabilities?

Sponsors and community driven