

vertiv

RQUEST™ - Regulatory Reporting

An introduction to **Regulatory Reporting Suite**



Real-time configuration

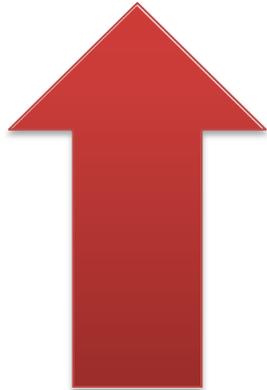
- Asset Class
- Rules, Formulae and Cells

Real-time trace-back

- Results to sources
- Data Lineage

Hierarchical Rule Definition Interface
Reports Metadata for Rule-Cell association

Rapidly changing regulatory requirements, limitation of organically grown systems and need for cross-functional warrants a configurable, rule-driven system with extensive auditing support



Rapidly changing Regulatory requirements

- Limitation of organically grown systems
- Need for cross functional systems



Increasing audit controls

- Visible calculations
- Trace-back



Increasing calculation complexity

- Increasing data granularity – contract level
- Time-series analysis

System Limitations

- Adhoc systems (Excel/Access/Scripts)
- Data Integration Challenges
- Rules hard-coded into the system

Implementation Challenges

- Drill back to source not possible
- Metadata and structural integrity missing
- Reconciliation needs to be done manually

Increasing Complexity

- Increasing calculation complexity and data volumes
- Several key attributes are dropped in aggregation

REQUEST

Regulatory Reporting

Financial
Rules Manager

Reconciliation
Engine

Report Metadata

Data Integrator

Hierarchy
Manager

Real-time Traceability Data model

- **Financial Rules Manager**
 - Two step transformation from transactions to report
 - Built in Statistical functions and time-series analysis
- **Reconciliation Engine**
 - Ability to bring in multiple datasets to same dimensionality
- **Report Metadata**
 - Ensures rules and results tie-up
- **Data Integrator**
 - Consolidates data silos
 - Supports XBRL

Solution approach delivers data model and business rules out of the box

Data Management

- Centralized Reference Data
- Strong and robust data model - End to end functionality
- Complete transaction data with no loss of granularity – Transparency issue
- Unification – Bringing data from various sources together
- Complete historical and market data to support risk calculation – High volume/Scalability
- Data archival and retrieval

Business Rules

- Configuration Driven – Changing a rule should not involve code change
- Transparent and Auditable
- Drill back capability for data and Lineage
- Performance Efficient
- Minimal Data Trucking
- Historical Rule Definition Maintenance

Calculations

- High performance
- Tightly integrated with underlying data model
- Support simulations
- Support stress testing
- Support complex mathematical functions

Traditional Rule Manager

- Good for transactional processing
- Good for low volumes of data
- Good for decision tree flow format with less nesting complexity

**H
A
S**

Limitation

- Cannot meet volume needs
- Does not typically access data sources or populate data targets directly
- Traceability / Drill back is expensive – consume storage
- Re-execution of rule required for drill back

**H
E
N
C
E**

Vertiv's Financial Rules Manager

- Set Operations can meet volume requirements
- Traceability / Drill back is query based instead of storing results
- Metadata approach maintains cell to rule linkage for drill back

Metadata driven approach makes it easy for on boarding new reports

Two step rule transformation approach brings modularity to process

Extensive support of finance functions ensure good coverage of the requirements

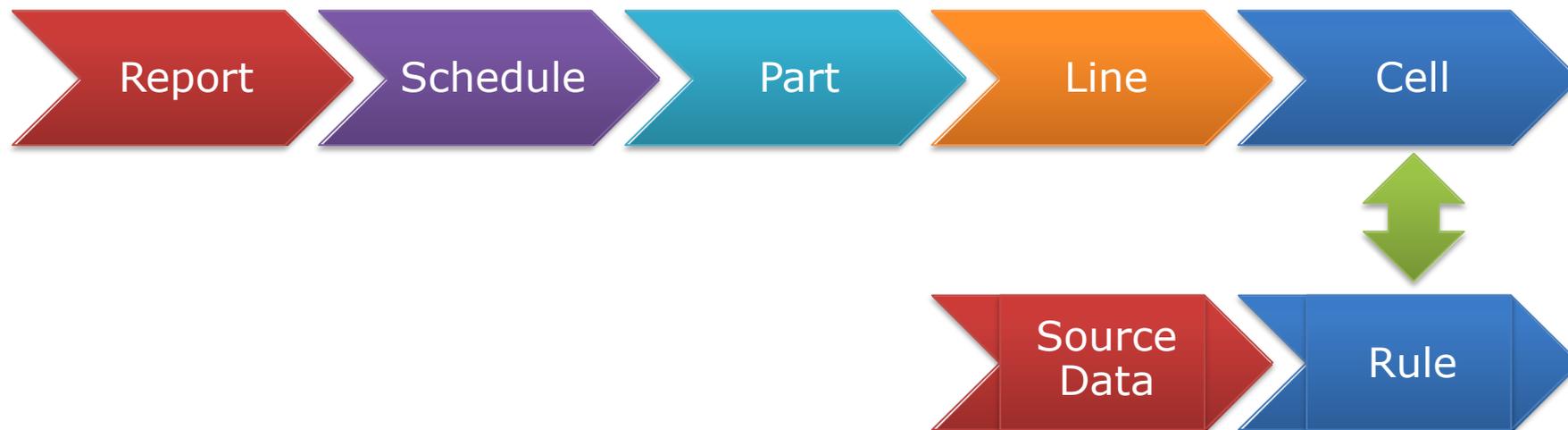
Inbuilt traceability and Reconciliation features ensure migration of rules and debugging easy

Set based operations can meet volume requirements

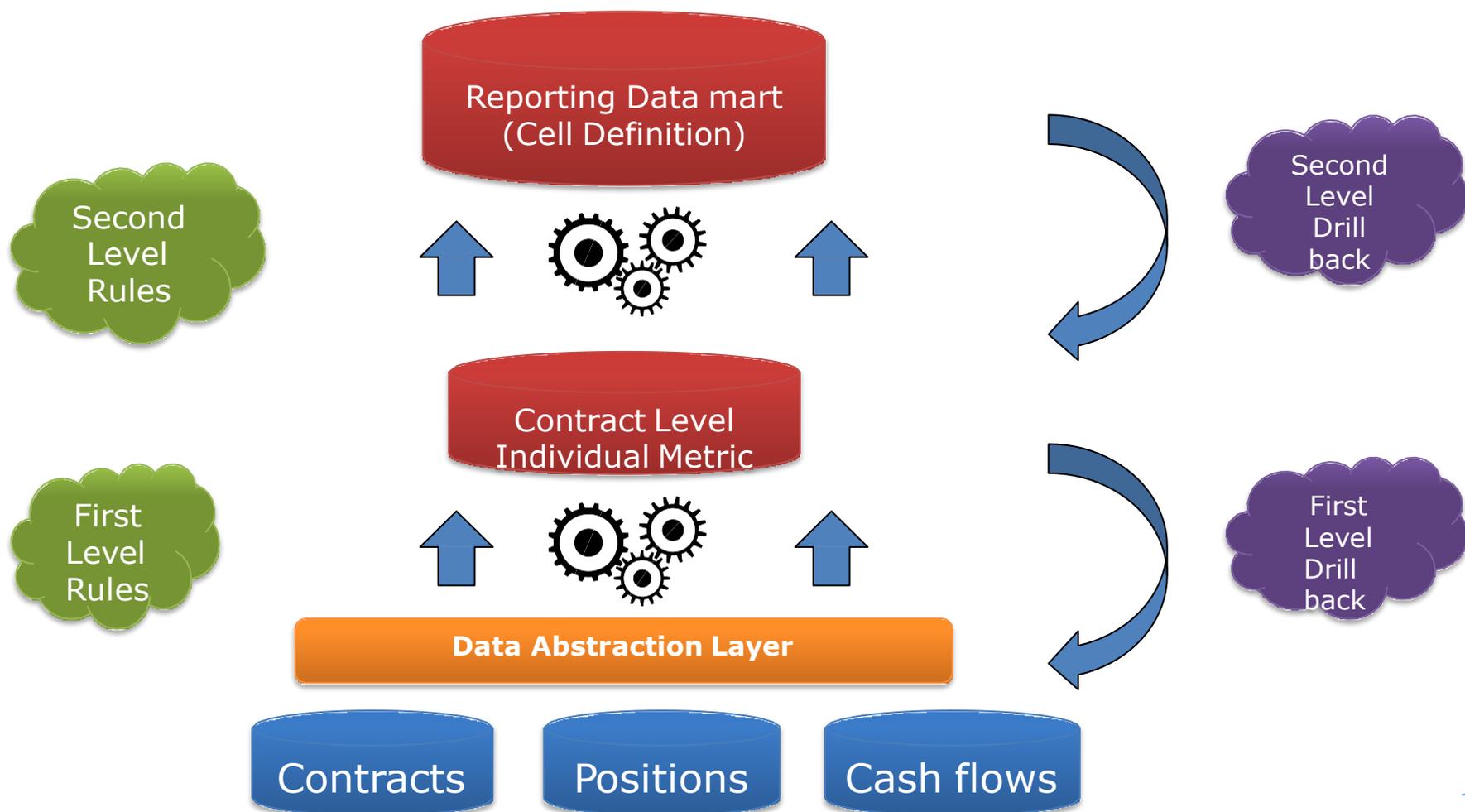
Traceability / Drill back is query based instead of storing results

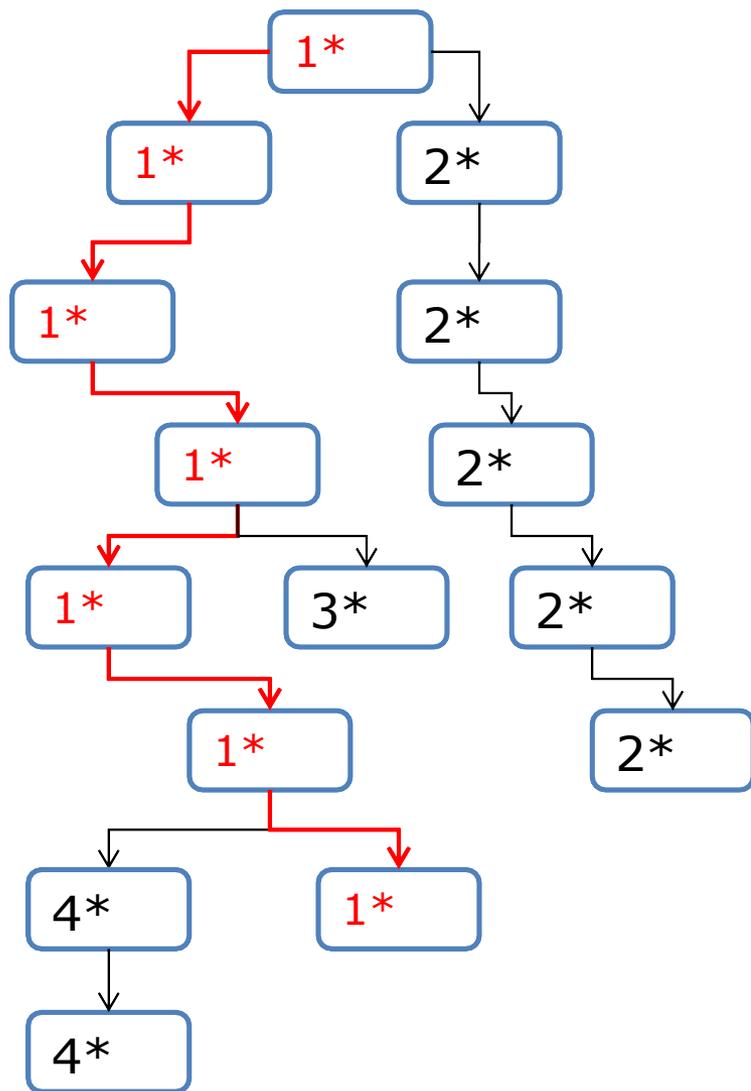
Metadata approach maintains cell to rule linkage for drill back

- ❑ Report metadata links Report – Schedule – Part – Line - Cell
- ❑ This allows marrying a regulatory rule to a report cell
- ❑ Rule to Cell association allows trace back from cell to source
- ❑ Report metadata also ensures if all the line items are being reported
- ❑ On boarding new regulatory report is configurable by business user and fast



- ❑ Minimized trucking of data ensures scalability
- ❑ Two stage approach for regulatory reporting ensures minimized data movement and enhanced performance





- Data lineage
 - Transformation **PATH** from source to target is always maintained

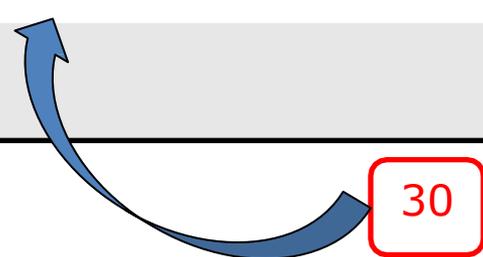
- Drill back
 - Intermediate **VALUES** at various stage are also maintained from source to target Identify sources of potential liquidity strain

- Ensures complete traceability and lineage for reconciliation and auditing requirements

Robust Adjustment Module

- Adjustments can be made at various stages of report gen
- Adjustments can be applied to other reports
- Adjustments are version controlled and audit trail generated
- Updating transactions supported for privileged users

Contractual	Spot	Open maturity	< 1wk	1wk -1m	1m-3m	3m-6m	6m-12m	>12m
Total Assets	110	20	10	30	55			
Total Liabilities	70	30	10			200	250	400
Off Balance Sheet Liquidity	20							



Contractual	Spot	Open maturity	< 1wk	1wk -1m	1m-3m	3m-6m	6m-12m	>12m
Total Assets	110	20	10					
Total Liabilities	70	30	20					
Off Balance Sheet Liquidity Mismatch	20							
Mismatch	-20	10	10					
Mismatch as % of total Assets	-18%	50%	100%					

- Dimension rich data mart supports management reporting
- Time-series analysis and financial functions supported

- ❑ Robust data integrator consolidates data silos
- ❑ Reconciliation module delivered for quick data checks
- ❑ Reports coverage for FFIEC 001-009, FR 2314 reports
- ❑ XBRL support for electronic filing
- ❑ Utilities like Hierarchy manager, Process Scheduler and Entitlements module
- ❑ User friendly features like copying and cloning of rules

- ❑ Regulatory Reporting warrants:
 - ❑ Contract Level Data capture from several source systems
 - ❑ Metric Level Processing
 - ❑ Data Aggregations with drill down support
 - ❑ Expandable for future requirements

- ❑ Ideal solution should offer
 - ❑ Massive Data Handling
 - ❑ Rules Drill back and Audit Tracing
 - ❑ Adjustments functionality

- ❑ Vertiv's RQUEST provides:
 - ❑ Real-time configuration of rules and reports
 - ❑ Real-time trace-back and lineage support
 - ❑ Robust stress testing support

Appendix

Dimension	Traditional Rule Engines	Vertiv RQUEST
Methodology	Processed one row at a time - slow	Many rows can be processed at a time
Execution	Java/C++ Based	SQL, CCL based
Processing Mechanism	Out of procedure <ul style="list-style-type: none"> • Longer Times (Latency) • Resource intensive 	In place data processing <ul style="list-style-type: none"> • No data hops • Faster processing
Rule Storage	File based	DB based
Suitability Scenarios	Broadly varying rules <ul style="list-style-type: none"> • Retail Industry • Insurance Sector 	Rules of similar nature <ul style="list-style-type: none"> • Aggregations • Summaries • Roll-ups
Processing Capacity	Few hundreds row transformations / sec	500K to 1 Million row transformations / sec

Dimension	Traditional Rule Engines	Vertiv RQUEST
Infrastructure Scalability	Low-medium <ul style="list-style-type: none"> • Clustering • Grid / Cloud 	Highly Scalable <ul style="list-style-type: none"> • Vertical • Horizontal
Processing choice	Possible Limitations	Diverse <ul style="list-style-type: none"> • Real-time (CEP) • Intra-day (Cache) • End-of-day (DB)
Utilities	Standard <ul style="list-style-type: none"> • Scheduler • Entitlements • Design/Debug and Production modes 	Standard <ul style="list-style-type: none"> • Scheduler • Entitlements • Design/Debug and Production modes Advanced <ul style="list-style-type: none"> • Tree Manager • Business Object Repository

❑ Definition Phase

- ❑ Define Liquidity Reporting Requirements
- ❑ Define and Build Liquidity Reporting Business Case
- ❑ Define solution architecture
- ❑ Define implementation delivery architecture
- ❑ Define Liquidity Reporting data architecture
- ❑ Define Integration architecture
- ❑ Define technical architecture
- ❑ Define test approach

❑ Design/Development Phase

- ❑ Design Liquidity Reporting Requirements Mapping
- ❑ Conduct Fit Gap Analysis
- ❑ Design Complete Solution Including Gaps
- ❑ Design information delivery solution
- ❑ Build and unit test information delivery solution
- ❑ Design and build data stores
- ❑ Design Integration solution
- ❑ Build and unit test Integration solution
- ❑ Develop test cases (integration, system and UAT)

❑ Testing Phase

- ❑ Execute integration test cycle
- ❑ Execute system test cycle
- ❑ Execute UAT test cycle
- ❑ Dry run with External organizations FSA, and ECB

❑ Deployment Phase

- ❑ Migrate code and Data to Production instance
- ❑ Configure and Deliver Production processes
- ❑ Transfer control over the system to Operational and Business Teams
- ❑ Prepare for Product Readiness
- ❑ Integrate and Activate DR Instance
- ❑ Move to Production

vertiv