



# IBM Almaden Research Center

Laura Haas

IBM Distinguished Engineer  
Director, Computer Science

Beauty and the Beast:

The Theory and Practice  
of Information Integration

Bertinoro

September 30, 2007





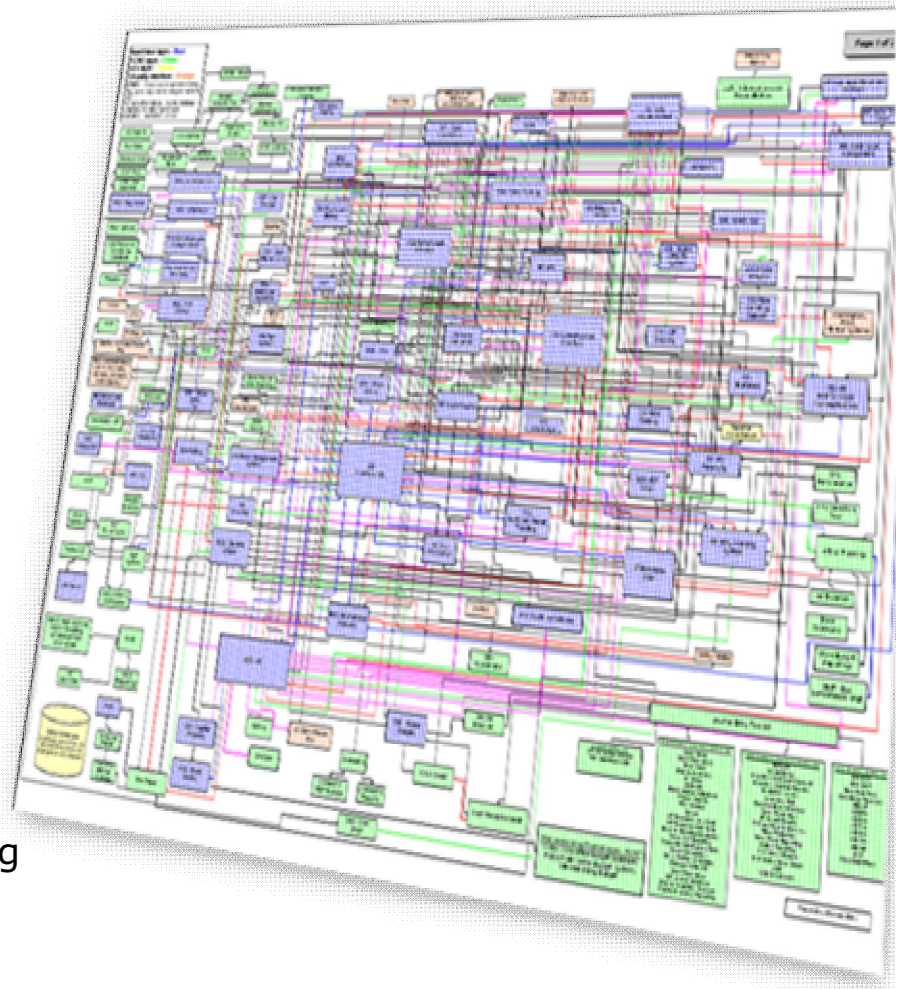
# Agenda

- The Beast
  - What makes integration hard?
  - A customer success story
- Taming the Beast
  - The state of the art in commercial integration technology
  - A simple integration scenario
  - The beast remains
- Breaking the Spell?
  - Raising the level of abstraction
  - Increasing automation



# What Makes Integration Hard?

- Diverse sources and types of information
  - Different models and operations
  - Different interfaces
- Limited knowledge about the information
  - Where is it from? How was it derived? How related to other information?
  - Complex environments the norm
  - Much of the knowledge is only in people's heads
- Overlapping and incomplete information
  - Choice of source(s)
  - Entity resolution
  - Reconciliation of information
- Solution requirements may vary
  - Is performance or availability (or both) important?
  - How current must the data be? How accurate?
  - Are there limits on storage space? Processing power?
  - What policies must be respected?

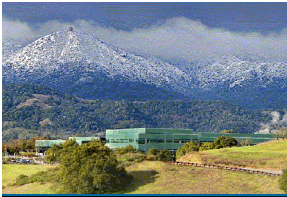




# Integration is a process

- Understand
  - What data is available
  - Important properties and values
  - Meaning or intent
- Standardize
  - Schema, field level, terminology and abbreviations
  - How to identify information about the same object
  - How to handle missing or inconsistent values
- Specify
  - Choose an integration engine or engines
  - Produce the executable(s) for integration
- Execute (Integrate)
- Iterate!!





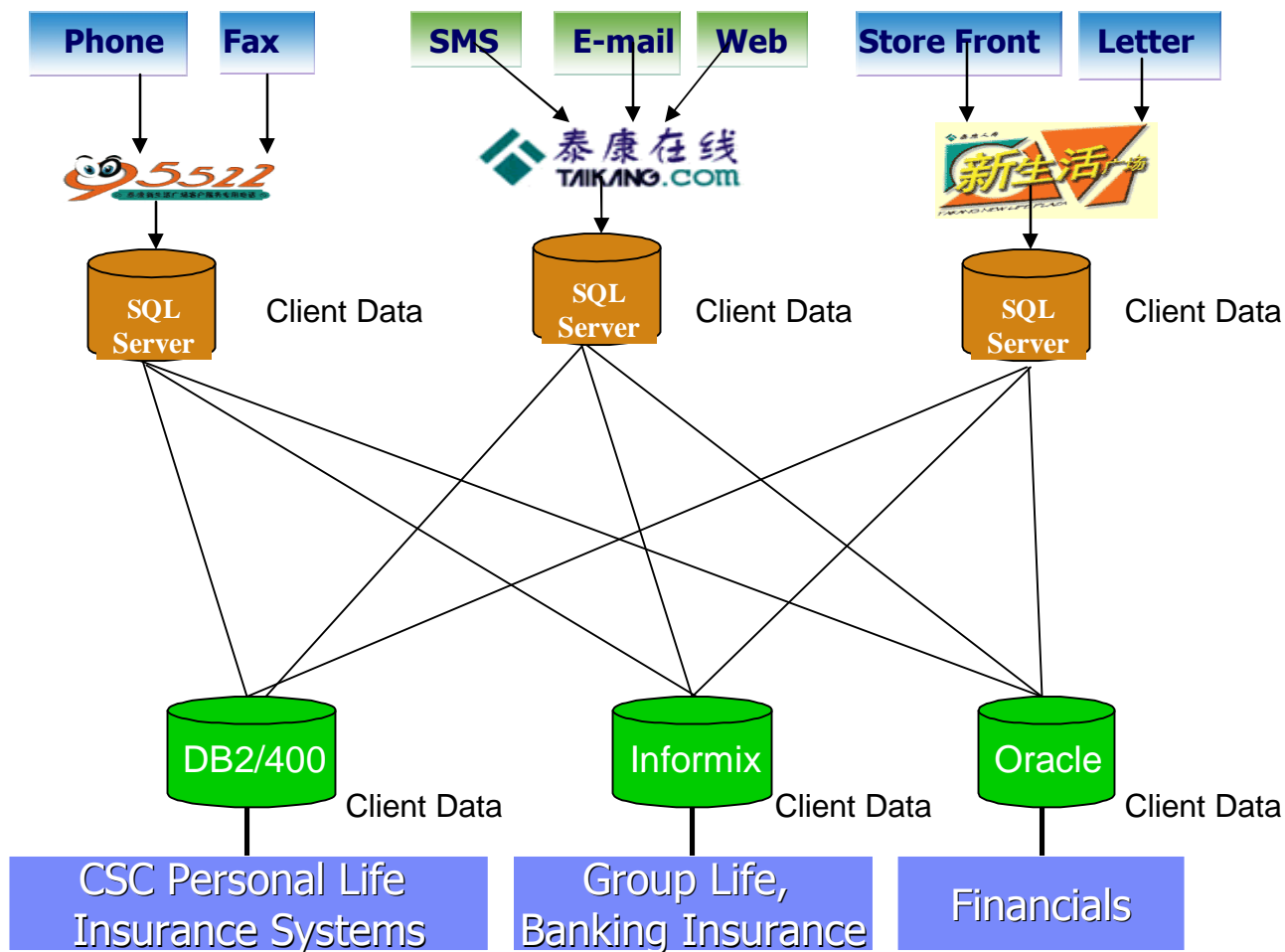
# Taikang Life Insurance: Getting a complete view of the customer

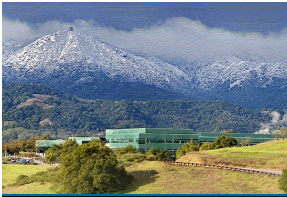
## Challenge

- Provide a clear picture of customers and associated services
- Capitalize on new business and growth opportunities with an up-to-the minute view of the organization

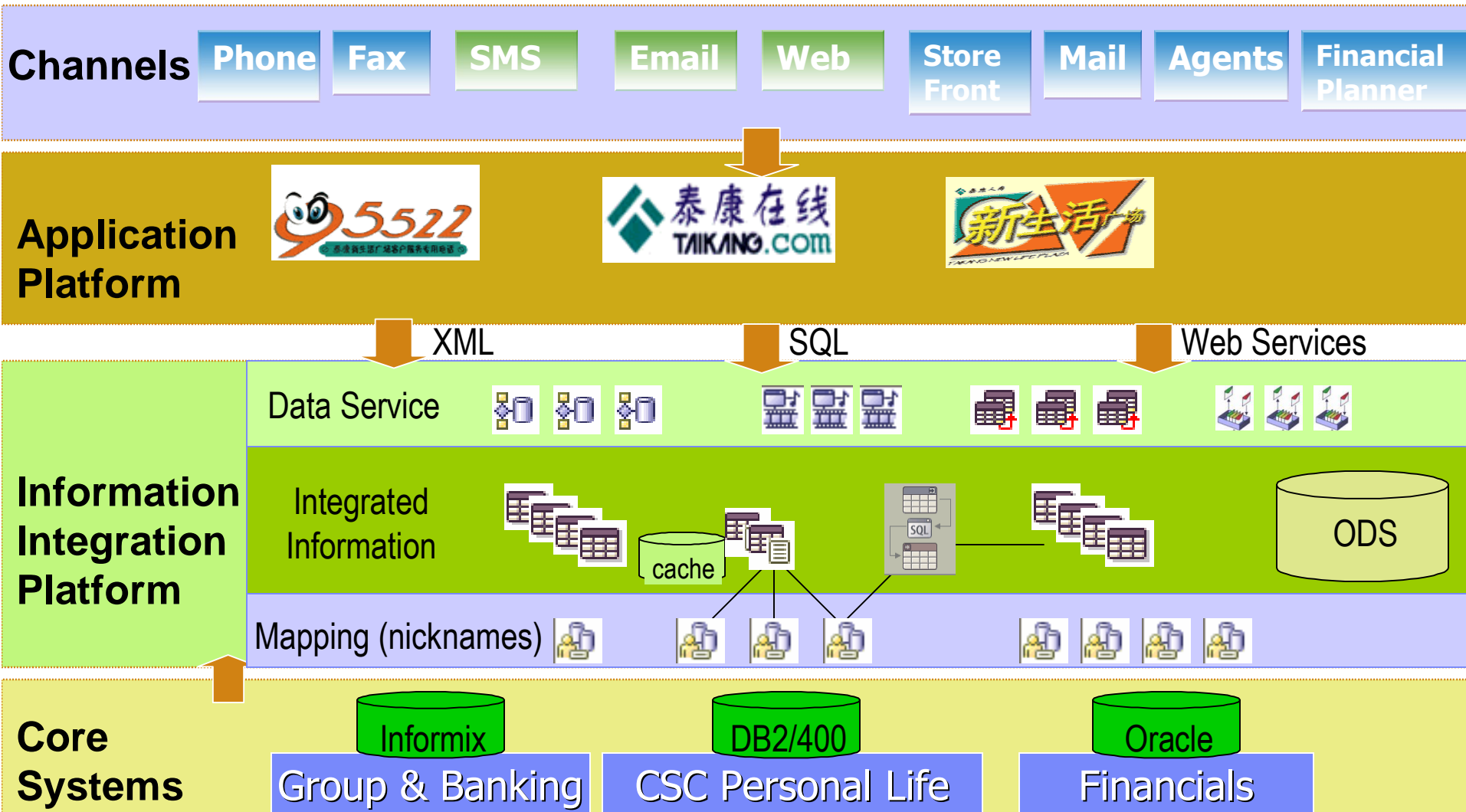
## Background

- 4th largest Chinese insurance company
- 8,000 employees, 150,000 agents
- 3.5 million customers
- Need information from DB2 UDB, Informix, Oracle, SQL Server, XML, e-mail, CRM and Portal





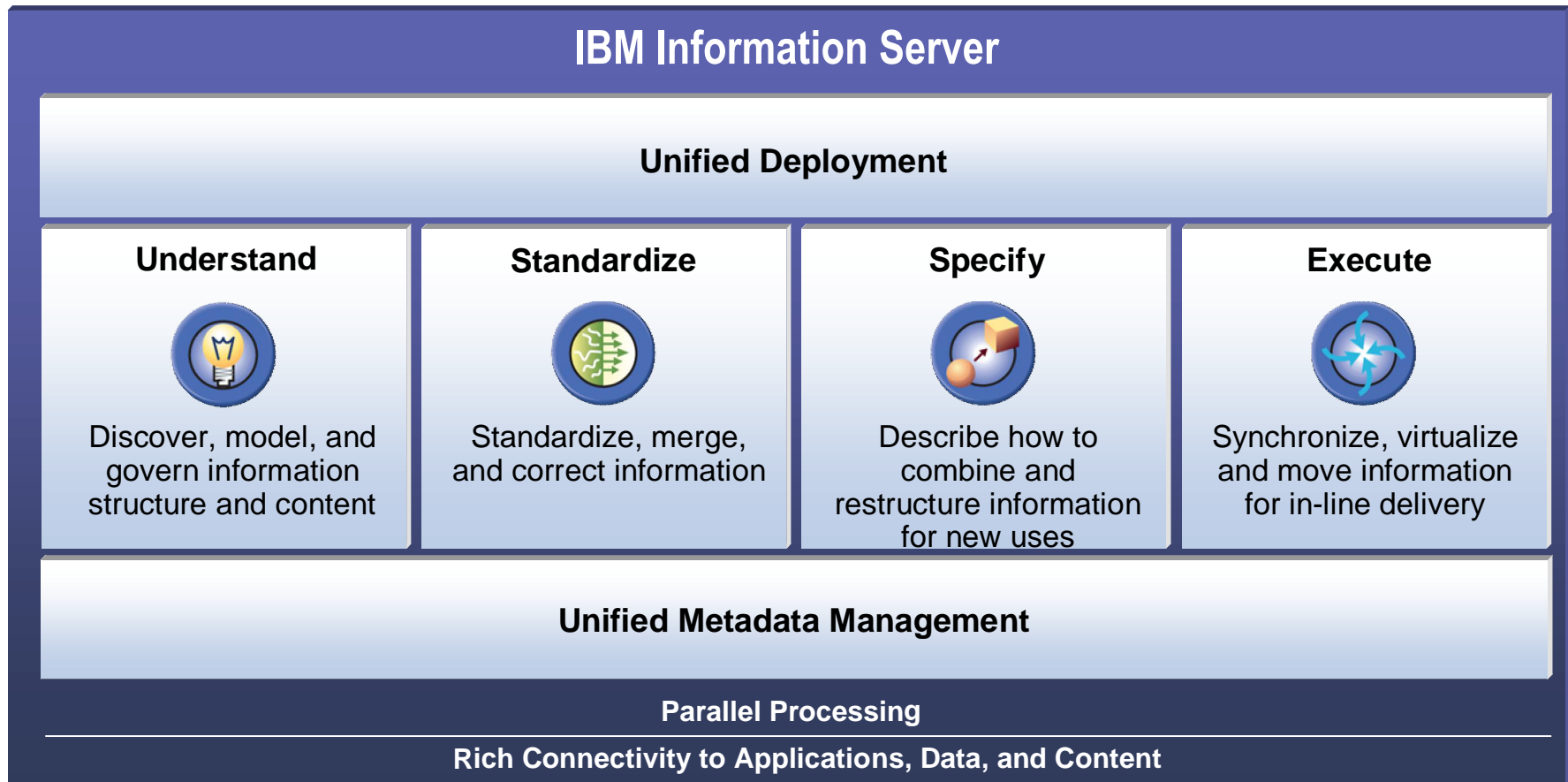
# Taikang Integrated Information Platform Architecture





# Marketing: IBM Information Server

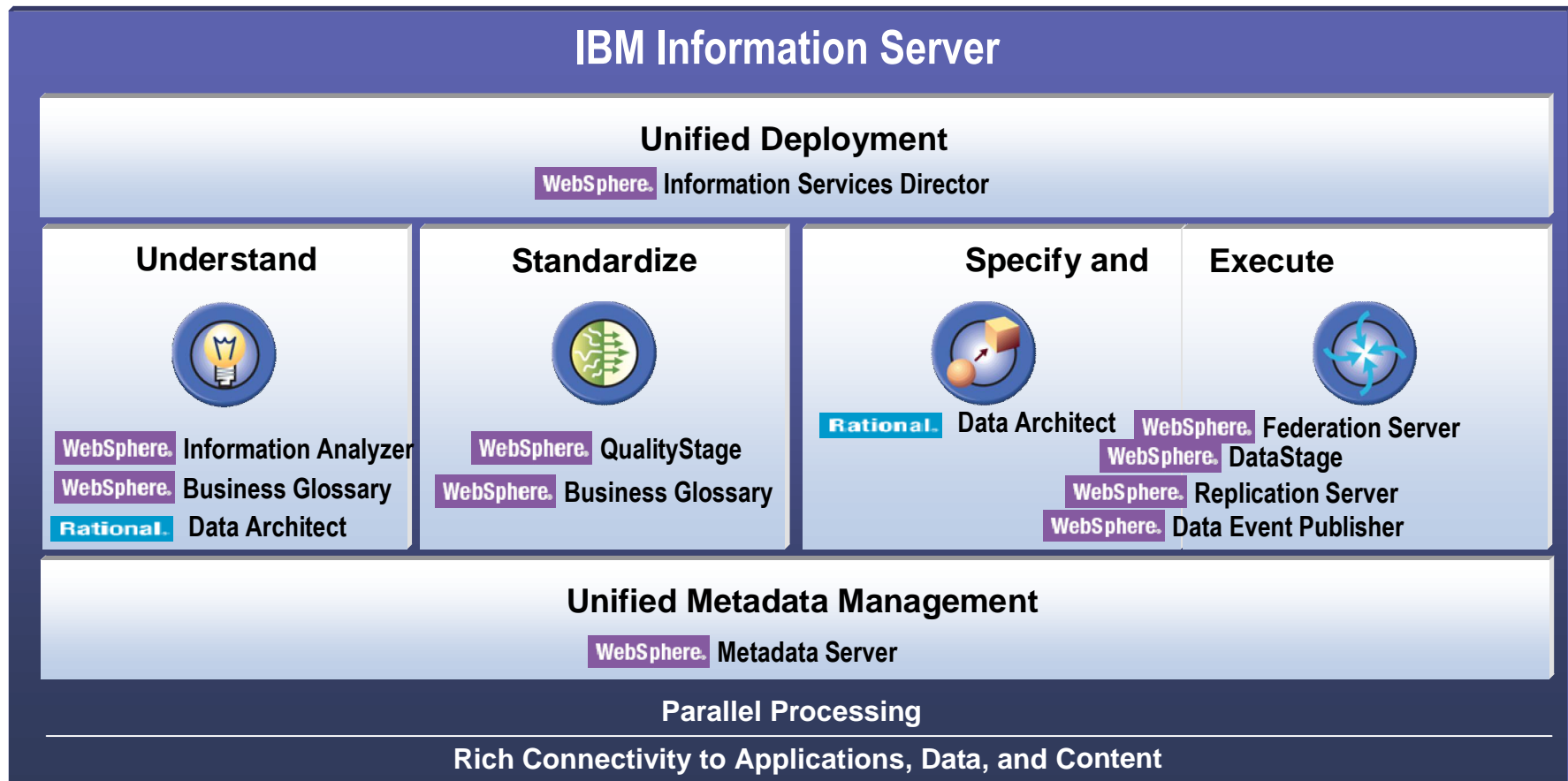
*Delivering information you can trust*



IBM is the acknowledged industry leader for vision and execution in information integration



# Reality: Lots of products, lots of choices

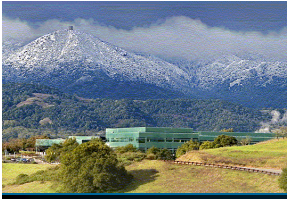


+ additional products for content federation, search, and special architectures or sources

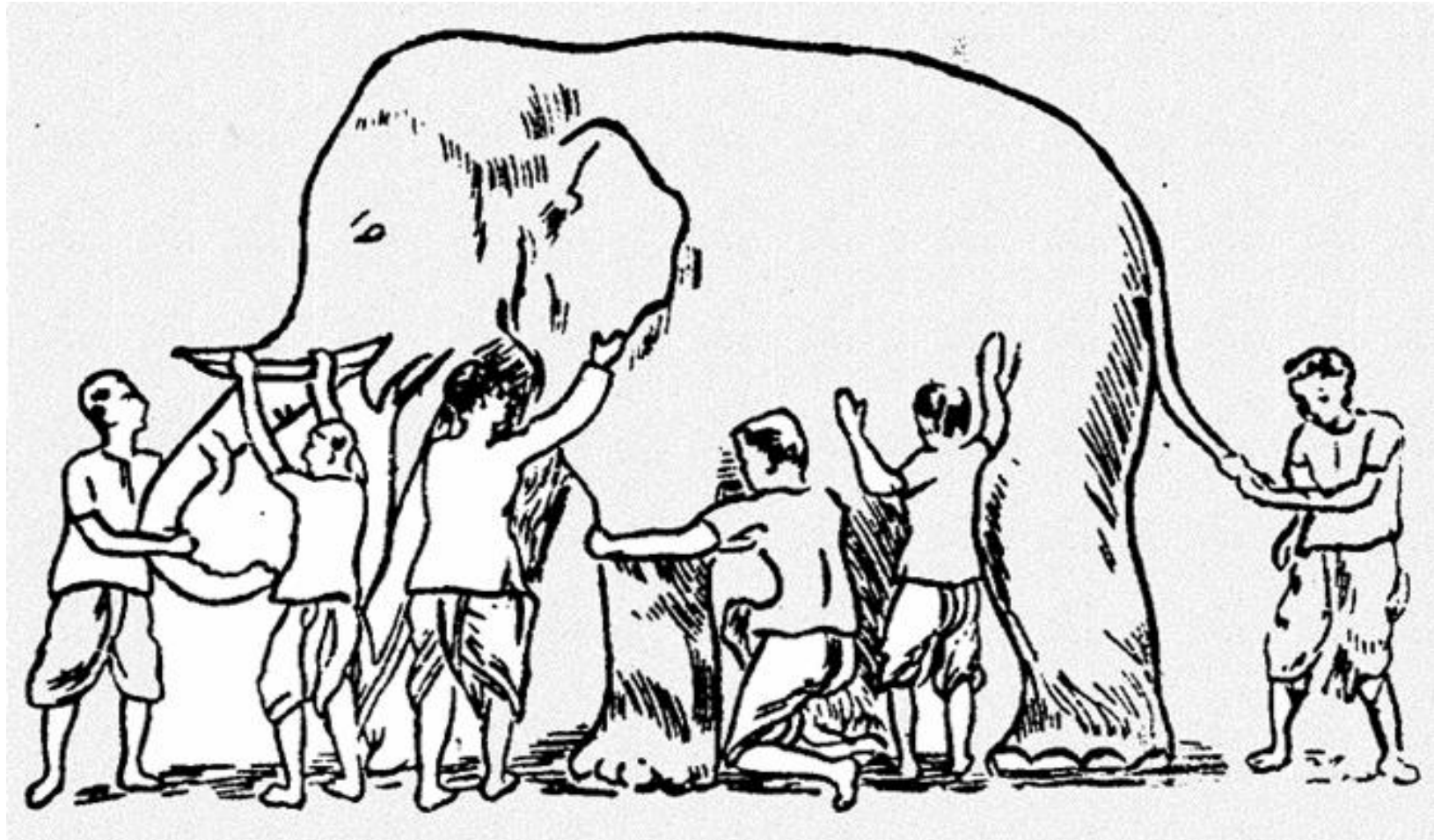
IBM is the acknowledged industry leader for vision and execution in information integration

- We can provide an end-to-end solution
- Otherwise, the problem is much worse – many vendors, mismatched products





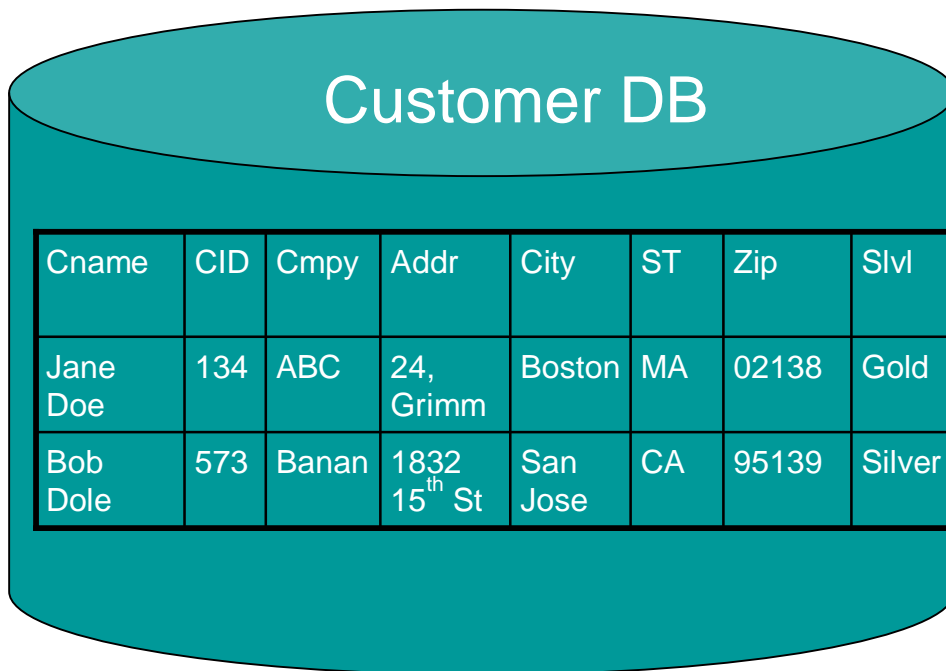
# Information Integration is Just \_\_\_\_\_



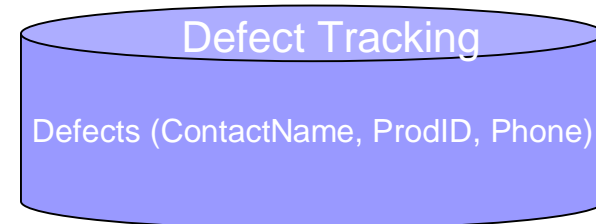
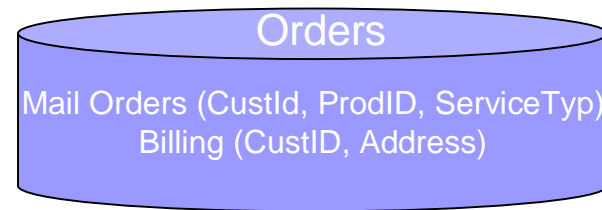


# Merging Chico and Grande

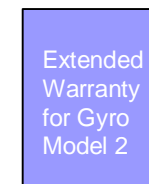
Grande Corporation



Chico



Warranties (Text)



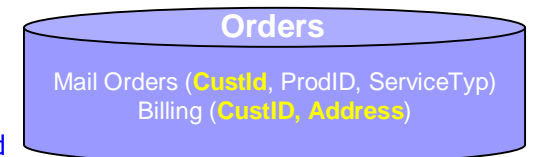


# Understanding What's There

- Where is the customer information?
- Assess the information quality
  - Is it complete? Any conflicts? Duplication?
- Understand statistical properties
  - Volumes? Important fields and distributions?
- Understand intent (semantics, meaning,...)
  - How are fields related? Any constraints?
- And so on...
- Available tools
  - Data explorers or profiling tools
  - Discovery engines
  - Enterprise catalogs
  - Modeling tools
  - Ontologies and business glossaries

ABC is both a mail order and web customer of Chico – and of Grande

Chico service levels relate to products' warranty and purchase of additional warranties, if any



Customer DB

Cname	CID	Cmpy	Addr	City	ST	Zip	Slvl
Jane Doe	134	ABC	2, Grimm	Boston	MA	02138	Gold
Bob Dole	573	Banan	1832 15 <sup>th</sup> St	San Jose	CA	95135	Silver

# RDA: Discover and visualize data sources

The screenshot displays the Eclipse Platform Data IDE interface. The main window shows a Physical Data Model (PDM) diagram with a central 'DEMO' node connected to three remote servers: 'FMWLOAN', 'SEEL9I.SVL.IBM.COM', and 'NW\_BANK'. Below these are various tables like 'FIRSTMIDW ESTLOANS', 'HR', 'COUNTRIES', 'NORTHWES TLOANS', 'AR', and 'ACCT\_INFO'. The left-hand 'Server Explorer' pane lists 'Available servers detected automatically', including 'Existing Remote Servers' and 'Discovered Remote Servers'. A 'Properties' window at the bottom shows details for a table named 'AR'.

**Available servers detected automatically**

**Visualize topology of data sources**

**Detailed view of properties**

Columns	Name	Primary Key	Datatype	Not Null	Generated	Default Value/Generate E..
Nicknames	AR_CR_RSK_RTG...	<input type="checkbox"/>	DATE	<input type="checkbox"/>	<input type="checkbox"/>	
Comments	AR_CR_RSK_RTG...	<input type="checkbox"/>	SMALLINT	<input type="checkbox"/>	<input type="checkbox"/>	
	AR_FNC_ST_DT	<input type="checkbox"/>	DATE	<input type="checkbox"/>	<input type="checkbox"/>	





# Information Analyzer: Looking at Data

The screenshot displays the Information Analyzer software interface. The main window is titled 'CUSTOMER CONSOLIDATION' and shows a 'Column Analysis' view for the 'Region' column. The interface includes a menu bar (File, Edit, View, Help), a toolbar, and a search box. The left sidebar contains a 'Repository' tree with folders for 'Shortcuts' and 'History', and a list of columns including Employee ID, Last Name, First Name, Birthdate, Hire Date, Address, City, Region, and Postal Code. The 'Region' column is selected and highlighted.

The main area shows the 'View Column Analysis' for 'Region'. The 'Frequency Distribution' tab is active, displaying the following summary statistics:

Total Rows	Data Class	Cardinality	Standard Deviation	Total Outliers
137	Code	57 (17%)	Code	57 (17%)

Below the summary, the 'Region Column' section shows options for 'Apply Filters' (checked), 'Full' (selected), 'Value Outliers', and 'Frequency Outliers'. A text input field contains 'Enter Text' and a numeric input field contains '20'.

The main data table displays the frequency distribution for the 'Region' column:

Data Value	Frequency		Value Flag	Data Type	Length	Format	Pattern	Transformation Value	Value		
	#	%							Definition	Source	Type
NULL	45	1.00	Valid	Char				Yes			
RECA	384	8.50	Valid	Char	4	AAAA		Yes			
REPL	769	17.09	Valid	Char	4	AAAA		Yes			
RECA	444	9.87	Valid	Char	4	AAAA		No			
RENO	252	5.66	Valid	Char	4	AAAA		Yes			
RENC	783	17.40	Valid	Char	4	AAAA		No			
R2*W	264	43.34	Valid	Char	4	AAAA		Yes			
R9UG	584	12.20	Valid	Char	4	AAAA		Yes			
RSPX	37	2.20	Valid	Char	4	AAAA		No			
RSLQ	87	8.39	Valid	Char	4	AAAA		No			
RSP2	45	21.94	Valid	Char	4	AAAA		No			

At the bottom right, there are buttons for 'Export Frequency Distribution', 'Close Analysis Detail', and 'Save All'.



# Information Analyzer: Looking at Data

The screenshot displays the IBM Information Analyzer interface. The main window is titled 'CUSTOMER CONSOLIDATION' and shows a 'Table 01' data source. The 'New Columns By' dropdown is set to 'Frequency Distribution'. The central chart shows the frequency distribution for various columns, with values ranging from 46 to 793. The right-hand pane, 'Table 01 Details', provides metadata for the table, including its name, data source, type, and review date.

**Table 01 Details**

Properties	
Name	Table 01
Data Source	Data Source A
Alias	
Keyword	
Type	Table
Type	Real
Entity Type	Primary
Data Group	Customer
Review Date	02/03/04
Reviewed By	Name Here

Columns	
Total	00000
Identifier	2
Code	5
Indicator	1
Large Object	0

Rows	
Primary Key	
Foreign Keys	



# Standardizing the Representation

- Determine how the information should be represented

- Target schema
- Field level data formats
- Abbreviations and terminology

**Grande address:** separate fields for street, city, state and zip. **Chico address:** a single string

Last name, first name or first then last?

Street = Str or St?

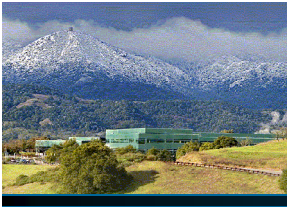
- Rules for repairing information

- Detecting overlap (object resolution)
- Dealing with missing data
- Handling inconsistencies

ABC has different customer ids, and phone numbers and addresses in Grande and Chico dbs

- Available tools

- Schema modeling
- Data cleansing
- Business glossaries, ontologies
- Triggers or other rule engines



# Create a standard model for loan information

**Associate a naming standard with model**

- Can be used to generate physical names
- As a thesaurus for relating schemas

**Import or visually create a standardized loan information model**

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows a tree view of data models including 'StandardizedLoanInfo-PM'.
- Physical Data Model:** Displays a diagram with tables and their attributes. Relationships are shown with lines and crow's foot notation symbols.
- Table Details:**
  - ARRANGEMENT:** Attributes include ARRANGEMENT\_ID, ARRANGEMENT\_TYPE, JE\_ID\_ON\_SOURCE\_SYSTEM, STRUCTURE\_DIFFICULTY\_LEVEL, STRUCTURE\_DELAY\_REASON, ICE\_SERVICE\_TERMINATION, ICE\_SERVICE\_UNTIL\_RANGE, ICE\_SERVICE\_REPAYMENT, INITIAL\_EXPOSURE\_RANGE, REINSTATEMENT\_STATUS, ICE\_SERVICE\_UTIL\_SEGMENT, VMENT\_PRIM\_PAYMENT, and ENT\_PAID\_NUMBER.
  - CREDIT\_RISK\_RATING:** Attributes include CREDIT\_RISK\_RATING\_ID, CREDIT\_RISK\_RATING, CREDIT\_RISK\_RATING\_DESCRIPTION, and CREDIT\_RISK\_RATING\_LEVEL.
  - CUSTOMER:** Attributes include CUSTOMER\_ID, UNIQUE\_ID\_ON\_SOURCE, PRIMARY\_RELATION\_TYPE, INDIVIDUAL\_LIFECYCLE\_T, CUSTOMER\_LIFECYCLE\_T, INDIVIDUAL\_AGE\_GROUP, INDIVIDUAL\_MARITAL\_STA, SOCIO\_ECONOMIC\_CATEG, INDIVIDUA, NAME, HOUSING, MONTHLY, CONTINUIT, EMPLOYER, INDIVIDUA, INDIVIDUA, INDIVIDUA, DESCRIP TI, COUNTRY\_, COUNTRY\_, COUNTRY\_, CUSTOMER, CUSTOMER, CUSTOMER, CUSTOMER, CUSTOMER, CUSTOMER, EFFECTIVE.
  - ACCOUNTING\_UNIT:** Attributes include ACCOUNTING\_UNIT\_ID, ACCOUNTING\_UNIT\_TYPE, RRANGEMENT\_ID [FK], CUSTOMER\_ID [FK], NIT\_OF\_MEASURE, AST\_UPDATE\_DATE, and AST\_UPDATE\_TIME.
  - MEASUREMENT\_PERIOD:** Attributes include MEASUREMENT\_PERIOD\_ID, MEASUREMENT\_PERIOD, MEASUREMENT\_PERIOD\_TYPE, NAME, NUMBER\_OF\_DAYS, CALENDAR\_QUARTER, CALENDAR\_MONTH, and WEEK\_OF\_CALENDAR\_YEAR.
- Server Explorer:** Shows existing database connections like 'FMWLOAN [DB2 UDB V8]'.
- Properties Palette:** Shows properties for the selected table, such as 'derived' (false) and 'editable' (true).
- Table Reference:** A window titled 'Data - FirstMidWestLoans-Glossary.ndm' displays a table with columns: Name, Abbreviation, Alternative..., Type, Modifier, and Descriptor. It lists various terms like ARRANGEMENT, AVAILABLE, ACTIVITY, etc.



# QualityStage Designer

Match Designer - Specifications: EXDMS.TCP

Open Pass Get Pass Remove Pass Move Left Move Right Save All Passes Special Variable Properties Save Match Specifications Configuration Hiding Area

Match Type: Reference Match

Flow Link: E14001  
Reference Link: E14002

RefPass1 RefPass2 RefPass3

Match Pass: RefPass1

Match Pass Holding Area

This area holds Match Passes that are not included as part of the Match job. To add a Pass, press the Cut key and drag the Pass from the Match Pass area.

Match Commands

Blockings Columns: Match, Quality, Control, Export, Columns

Descriptions:

- BLOCK1
  - Type = ALPHERIC
  - Data Column = STREET
  - Reference Column = STREET\_Ref
- BLOCK2
  - Type = CHARACTER
  - Data Column = TYPE
  - Reference Column = TYPE\_Ref

Match Commands

Blockings Columns: Match, Quality, Control, Export, Columns

Descriptions:

- COMMAND1
  - Type = INT\_INTERVAL
  - Data Column = HQCUST
  - Reference Column = LPHCUST
  - Reference Column = HQCUST
  - Reference Column = LPHCUST
  - Reference Column = HQCUST
- COMMAND2
  - Type = INT\_INTERVAL
  - Data Column = HQCUST
  - Reference Column = HQCUST
  - Reference Column = HQCUST

Match Count: 1024    Global Count: 1000    Duplicate Count: 1000

Data: Total Records: 16

Export Data Selection

SeqID	Record Type	Weight	Match Command (COMMAND1)	Match Command (COMMAND2)	Blocking Column: BLOCK1	Blocking Column: BLOCK2
1	Match	4.00	34	34	M024	PL
2	Reference	4.00	90   000   000   000	C=00000000   C=00000000	M024	PL
3	Match	4.00	125	125	M024	ST
4	Reference	4.00	195   100   1000   100	C=00000000   C=00000000	M024	ST
5	Match	4.00	134	134	M024	ST
6	Reference	4.00	105   100   1000   100	C=00000000   C=00000000	M024	ST
7	Match	4.00	52	52	M024	ST
8	Reference	4.00	90   000   0000   000	C=00000000   C=00000000	M024	ST
9	Match	4.00	202	202	M024	ST
10	Reference	4.00	296   200   0000   000	E=00000000   H=00000000	M024	ST
11	Match	4.00	116	116	PLM	ST
12	Reference	4.00	140   100   0000   000	E=00000000   H=00000000	PLM	ST
13	Match	4.00	200	200	PLM	ST
14	Reference	4.00	296   200   0000   000	E=00000000   H=00000000	PLM	ST
15	Match	4.00	18	18	M024	ST
16	Reference	4.00	90   000   0000   000	C=00000000   C=00000000	M024	ST



# Specification and Execution

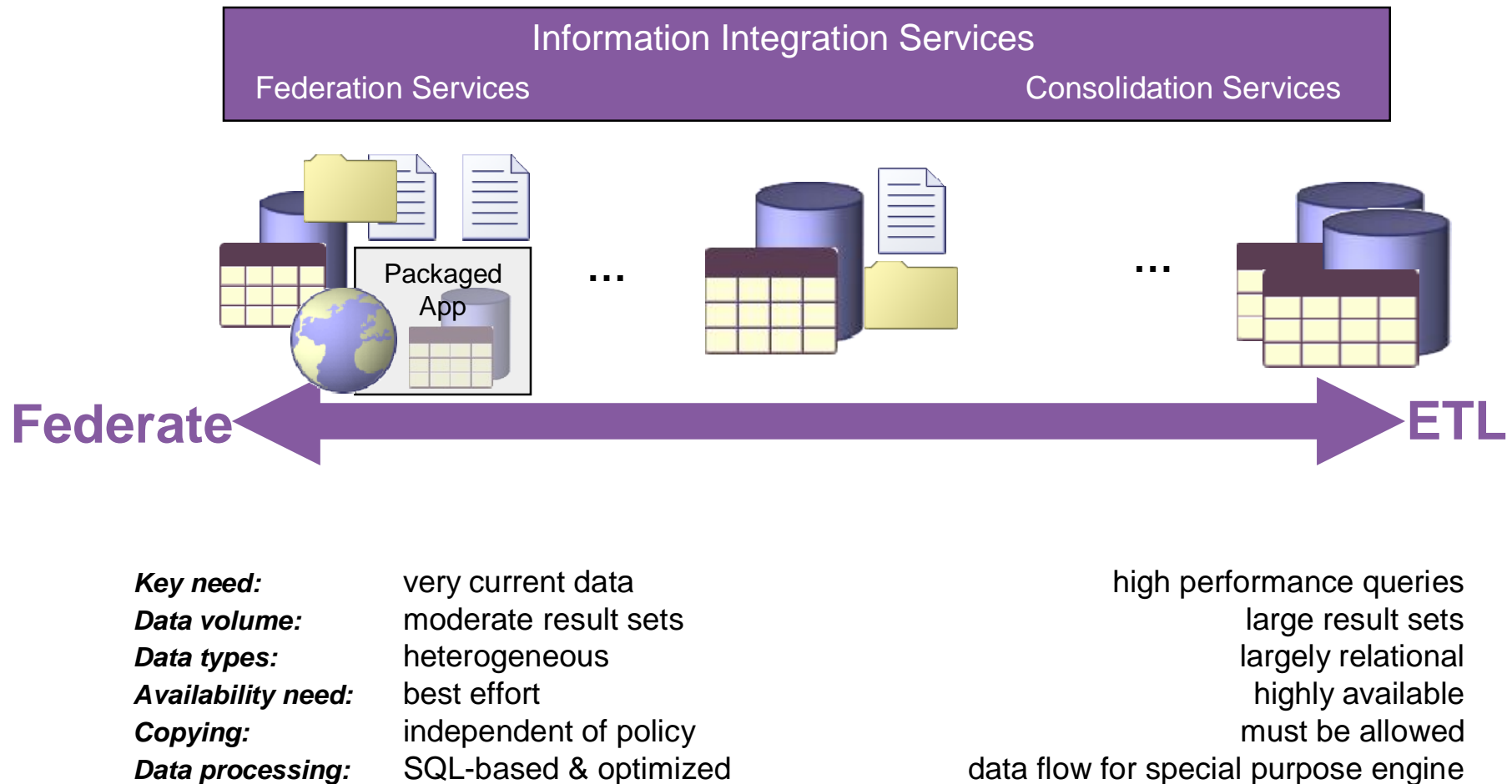
- What execution engine(s) should be used?
  - Materialize
    - Extract, Transform, Load (ETL)
    - Replication
  - Virtualize (Federate)
  - Search
- What input is needed to drive the execution engine of choice?
  - ETL: Dataflow graph or script
  - Replication: subscription definitions, e.g.
  - Federation: SQL queries
  - Search: Keywords
  - Plus, configuration of each engine (connectivity, tuning, ...)
- Available tools
  - Each engine usually provides some configuration tool
  - Query builders for SQL
  - Many engines... all different, few standards





# Choosing an Execution Engine: Example 1

## Materialize or Federate

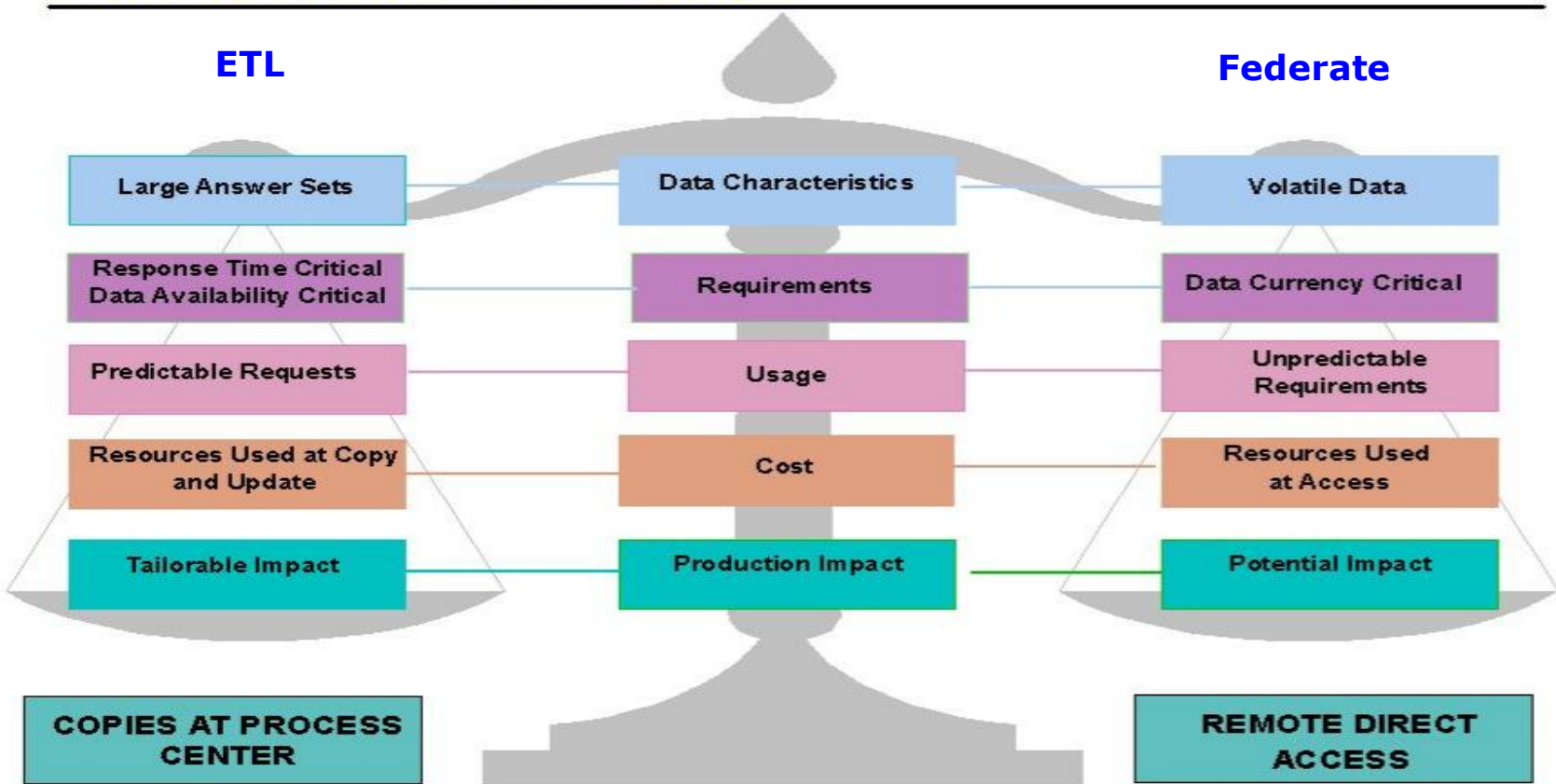




# Choosing an Execution Engine: Example 2

## Materialize or Federate

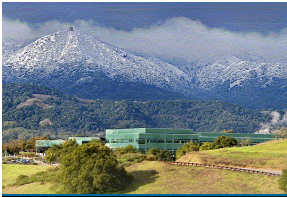
### *Copies or Remote Access Criteria*



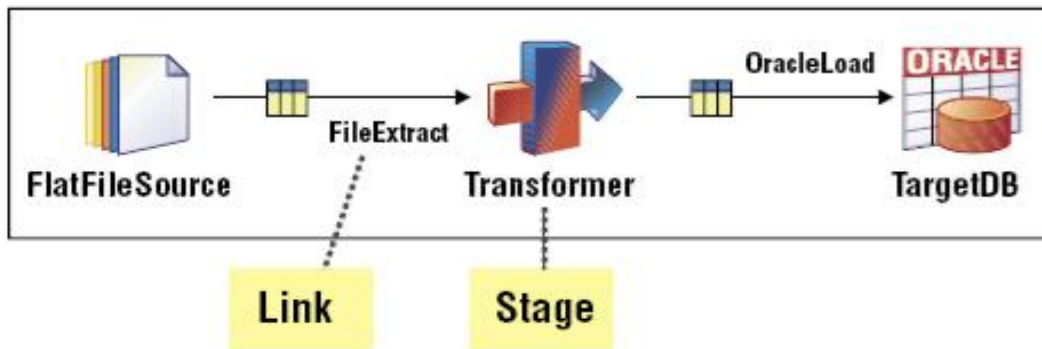
*IBM Software*

© IBM | 1998





# A Very Simple DataStage Job



The screenshot shows the **Palette** window in DataStage, which is used to select and drag components into a job design. The palette is organized into several categories:

- General**
- Data Quality**
- Database**
- Development/Debug**
- File**
- Processing**
  - Aggregator
  - Change Apply
  - Change Capture
  - Compare
  - Compress
  - Copy
  - Decode
  - Difference
  - Encode
  - Expand
  - External Filter
  - Filter
  - FTP Enterprise
  - Funnel
  - Generic
  - Join
  - Lookup
  - Merge
  - Modify
  - Remove Duplicates
  - Sort
  - Surrogate Key Generator
  - Switch
  - Transformer
  - Slowly Changing Dimension
- Real Time**
- Restructure**
- Favorites**

# Discover and relate existing schemas to standard

The screenshot displays the Eclipse Platform interface for a data model. The main workspace is divided into three panes: Source, Mappings, and Target. The Source pane shows the 'FMWLOAN.dbm' schema with various tables and columns. The Target pane shows the 'StandardizedLoanInfo-PM.dbm' schema with its corresponding tables and columns. Blue arrows indicate the mapping between columns in the source and target schemas. A green box labeled 'Existing schema' points to the Source pane, and another green box labeled 'Standard model' points to the Target pane. A third green box labeled 'Semantic discovery automatically relates schemas' points to the mapping arrows. A fourth green box labeled 'Annotate mapping with transformations and conditions' points to the 'Transformation' field in the Properties pane, which contains the SQL expression: `StandardizedLoanInfo.LOOKUP (FIRSTMIDWESTLOANS.MSR_PRI`. The Project Explorer on the left shows the project structure, and the Server Explorer on the bottom left shows existing database connections.

**Existing schema**

**Standard model**

**Semantic discovery automatically relates schemas**

- Schema and data-based
- Exploits glossary
- Reports relative ranking

**Annotate mapping with transformations and conditions**

Transformation: `StandardizedLoanInfo.LOOKUP (FIRSTMIDWESTLOANS.MSR_PRI`



# Relate and map existing schemas to standard

The screenshot displays the Eclipse Platform Data IDE interface. The main workspace is divided into three panes: Source, Mapping Groups, and Target. The Source pane shows a schema named 'FMWLOAN.dbm' containing tables 'FIRSTMIDWESTLOANS', 'AR', 'AR\_CR\_RSK\_PRFL', 'CL', 'CL\_SCM', 'IP', and 'MSR\_PRD'. The Target pane shows a schema named 'StandardizedLoanInfo-PM.dbm' containing tables 'StandardizedLoanInfo', 'ARRANGEMENT', 'MEASUREMENT\_PERIOD', 'CUSTOMER\_CREDIT\_RISK\_PRO', 'CREDIT\_RISK\_RATING', 'CUSTOMER', 'ACCOUNTING\_UNIT', and 'ACCOUNTING\_UNIT\_BALANCE'. A green arrow points from the 'AR' table in the Source to the 'CUSTOMER\_CREDIT\_RISK\_PRO' table in the Target. A second green arrow points from the 'AR\_CR\_RSK\_PRFL' table in the Source to the same 'CUSTOMER\_CREDIT\_RISK\_PRO' table in the Target. A third green arrow points from the 'AR' table in the Source to the 'AR\_IC' join in the Mapping Groups pane. The Properties pane at the bottom shows the 'Join' property with the expression: `FIRSTMIDWESTLOANS.AR.AR_ID = FIRSTMIDWESTLOANS.AR_CR_RSK_PRFL`. The Project Explorer on the left shows a tree view of the project 'RDA-Demo' with sub-projects for Data Models, Scripts, XML Schemas, and Mappings. The Server Explorer on the left shows a list of existing database connections.

*Visualize relationships at table level*

*Maps can be combined and composed*

*Discovers join conditions*



# Generate compliant view over existing schema

The screenshot shows the Eclipse Platform interface for a Data project named 'CreditRiskProfile.msl'. The 'Project Explorer' on the left shows a tree structure with 'Mappings' containing several mapping files. The 'Server Explorer' at the bottom left shows 'Existing Connections' including 'FMWLOAN [DB2 UDB V8.1]'. The main workspace displays a 'Mapping Groups' view with 'Source' and 'Target' columns. The source is 'FMWLOAN.dbm' containing 'FIRSTMIDWESTLOANS' with tables 'AR', 'AR\_CR\_RSK\_PRFL', 'CL', 'CL\_SCM', 'IP', and 'MSR\_PRD'. The target is 'StandardizedLoanInfo-PM.dbm' containing 'StandardizedLoanInfo' with tables 'ARRANGEMENT', 'MEASUREMENT\_PERIOD', 'CUSTOMER\_CREDIT\_RISK\_PRO', 'CREDIT\_RISK\_RATING', and 'CUSTOMER'. A green arrow indicates a mapping from 'AR\_CR\_RSK\_PRFL' to 'CUSTOMER\_CREDIT\_RISK\_PRO'. A 'Generation Wizard' dialog box is open, showing a 'Generation Summary' and the following SQL code:

```
CREATE VIEW GeneratedSchema.CUSTOMER_CREDIT_RISK_PROFILEView AS
SELECT S0.CST_ID AS CUSTOMER_ID,
       S1.MSR_PRD_ID AS MEASUREMENT_PERIOD_ID,
       FLOOR(AVG(S1.INR_CR_RSK_RTG_ID)) AS CREDIT_RISK_RATING_ID
FROM GeneratedSchema.ARNN S0,
     GeneratedSchema.AR_CR_RSK_PRFLNN S1
WHERE S0.AR_ID = S1.AR_ID
GROUP BY S0.CST_ID, S1.MSR_PRD_ID;
```

Below the SQL code, a table lists the generated objects:

Schema	Name	Type
GeneratedSchema	ARNN	Nickname
GeneratedSchema	AR_CR_RSK_PRFLNN	Nickname
GeneratedSchema	CUSTOMER_CREDIT_RISK_PROFI...	View

The 'Properties' view at the bottom shows 'Expression: FIRSTMIDWESTLOANS'. A green callout box with a white border contains the text: 'Generate a view that renames and restructures existing schema in terms of standard model'. The 'Generation Wizard' dialog has 'Back', 'Next >', 'Finish', and 'Cancel' buttons at the bottom.



# Customers Want Integration, Not Federate, Materialize, Search, ...

- Hard and dangerous choices
  - Too many products with too much functional overlap
  - Too hard to choose, have to choose too early
  - Too hard to switch among products
- Too hard to use to build effective solutions
  - Too many knobs, too much training to use well
  - Too stove-piped (hard or expensive to use in combination)
  - Too little easy life-cycle flow among products
- Time to delivered value is too long
  - Need consulting-services engagements to survive
  - No way to kick-start the process
- Integration never ends
  - Integration usually starts with a project or two
  - Future projects should extend, reuse – avoid duplicate work



# We Need to Simplify the Process

- Integration is the new data access
  - Applications need information that comes from multiple sources
  - Thus, writing applications that access data => integration
  - But integration is too difficult
- We need to raise the level of abstraction
  - Use “semantics” (more information)
  - Generate configuration and execution artifacts automatically
- Goal: non-procedural integration
  1. Automated understanding and standardization
  2. Complete, high level specification
    - **Logical:** what information is wanted
    - **Practical:** what qualities must the solution have
  3. Tools to turn the high level description into actionable specifics
  4. Powerful engine (or engines?) to do the specified integration
- Research issues for both systems and theory communities

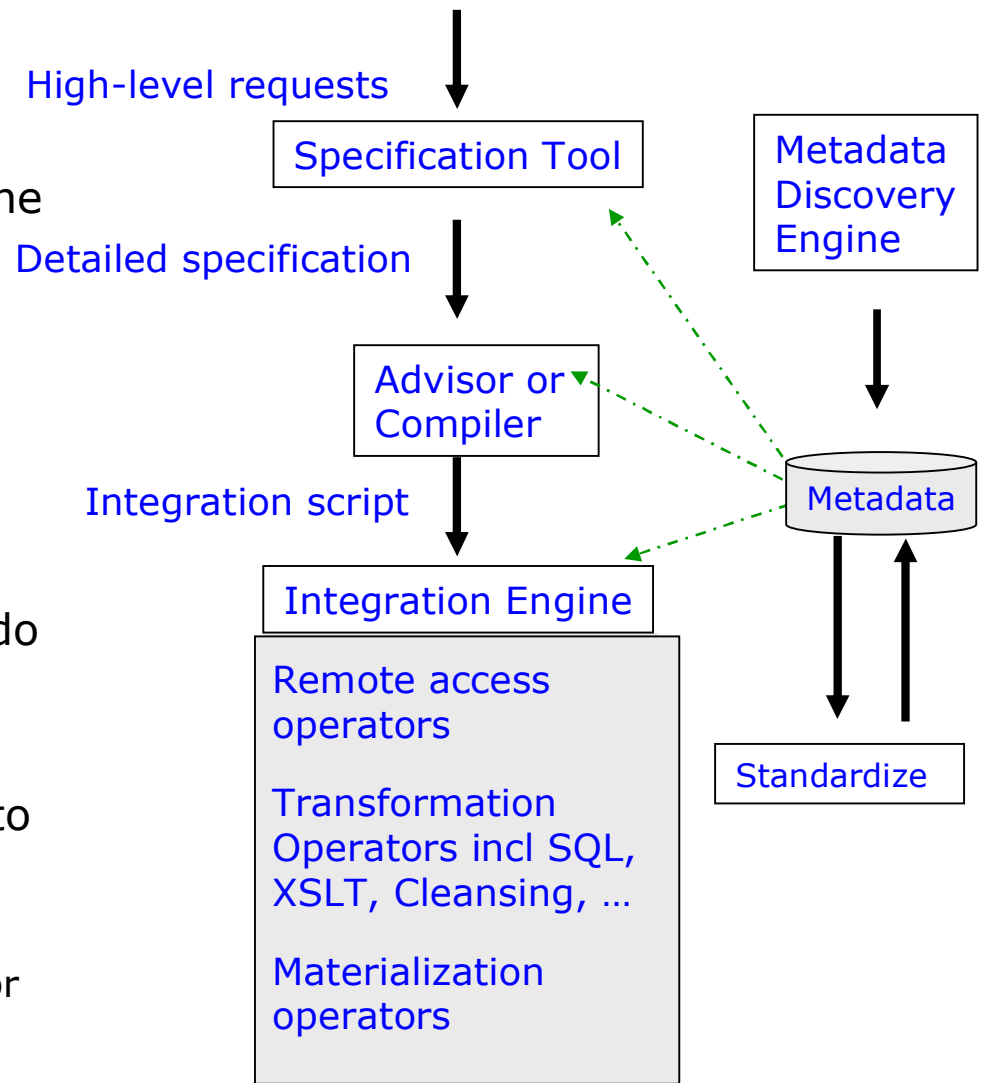


# Some “formal” issues

- Creating a complete picture of information integration
  - What are the fundamental operations for information integration?
  - Can we provide a theoretical basis for the information integration process?
  - Can data integration and data exchange even be viewed as different implementations of the same task?
- **Hypothesis:** there is a set of requirements (the desiderata) that must be represented and reasoned about as part of any complete integration solution.
  - What are the critical solution desiderata?
  - How do they relate to the integration techniques needed?
  - Can we represent them formally? Which? How?
  - If they were formally represented, what could we learn? Do?
- Will a formal theory help create a single integration engine for all integration tasks?
  - Is this a worthy goal? (It sounds simpler...)
  - What is the analog of the relational calculus for integration?
  - How close can we come to fully automating integration?

# Some systems issues

- What software do we need?
  - Automate understanding and standardization
  - Specify the solution desiderata
  - Integration Advisor to suggest the engine?
  - “Compiler” to generate the integration “script”?
  - How much can we automate?
- How should the integration engine be structured?
  - A “blade” approach: Specialized engines as needed to federate, do bulk transforms, search, ...?
  - Or a “super-engine” that does it all?
  - How to make it simple yet able to meet a range of requirements?
    - Parallel infrastructure for performance, scalability?
    - Services-oriented architecture for flexibility?







# Saying What Is Desired: Desiderata

- **Logical:** How to specify what information is needed?
  - Naming?
    - “Information about customers”, e.g. (not “cname from client”)
  - Form of the request?
    - From precise query to imprecise search
    - Language or a set of operators or ?
  - Metadata?
- **Practical:** What other properties of the solution are desired?
  - Qualities of service
    - Performance, availability, scalability, etc
  - Qualities of data
    - Currency, completeness, accuracy, etc
  - Physical constraints that must be met
    - Available storage, processing power, floor space, ...
  - Policies that must be obeyed
    - Privacy, security, cost sharing, legal, ...
  - Which of these are critical for determining integration techniques?
  - How can they be modeled?



# Summary: Can Beauty Tame the Beast?

- Information integration is a beast
  - Hard and pervasive
  - A multi-stage process
  - Many (incompatible) choices of technology for each phase
  - Little guidance on how to choose
- Research has looked only at aspects of the big picture
  - Progress on some fundamental issues
  - Integration technologies for pieces of the problem
  - There are many open problems
- The *desiderata* are key
  - Logical: target schema, constraints
  - Practical: Qualities of service and data, physical constraints, policies
  - No precise decision rules today
- Can we automate more of the integration process?
  - Raise the level of abstraction
  - Derive appropriate decision rules
  - Use them to create nonprocedural integration



# Thanks

- Phokion Kolaitis, Ron Fagin, Lucian Popa
- The Clio team
  - Renee Miller, Mauricio Hernandez, Lucian Popa, Howard Ho...
- IBM Information Integration Solutions development team
  - Lee Scheffler, Mike Beckerle...
  - Technical enablement and strategy teams
- The Garlic team
  - Peter Schwarz, Mary Roth...
- And many colleagues world-wide