MANCHESTER 1824
The University of Manchester

Evolutionary Design: Data Evolution - Challenges

Andy Carpenter

School of Computer Science

(Andy.Carpenter@manchester.ac.uk)

Elements these slides come from Sommmerville, author of “Software Engineering", and are copyright Sommerville

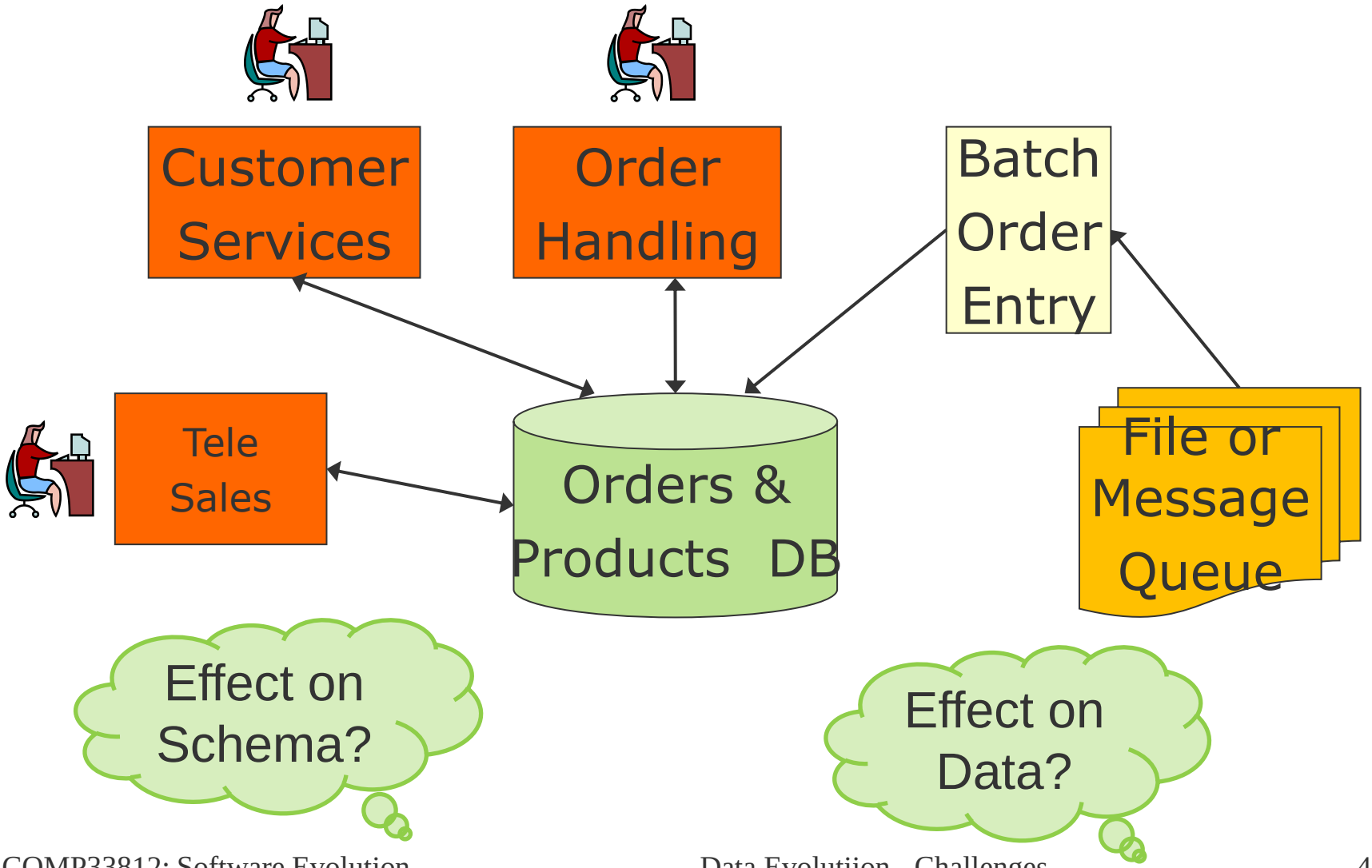
Data-Intensive Systems

- Concerned with the storage and manipulation of large amounts of data
  - Usually persistent
- Includes majority of modern business, government and scientific systems
- Evolution of such systems is difficult
  - For all the reasons we have discussed so far
  - Plus some additional data-specific challenges

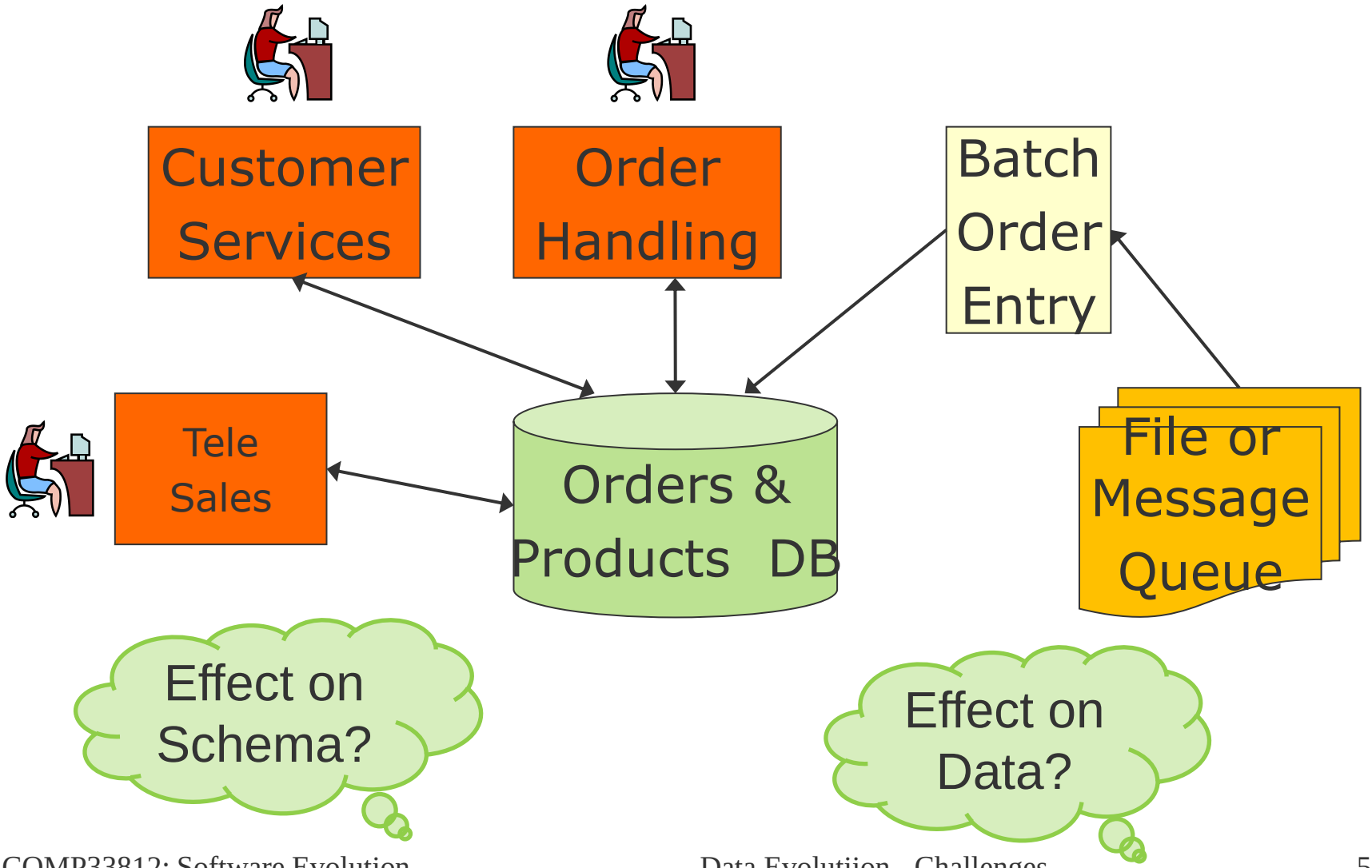
Data as an Asset

- Data as long-term organisational memory
  - customer understanding/management
  - scenario modelling, data mining
- Data is “declarative”
  - can support new applications, including applications not envisaged when the data was first captured
- Schema often models core domain concepts
- Raw data may be even more revealing than the DB structure

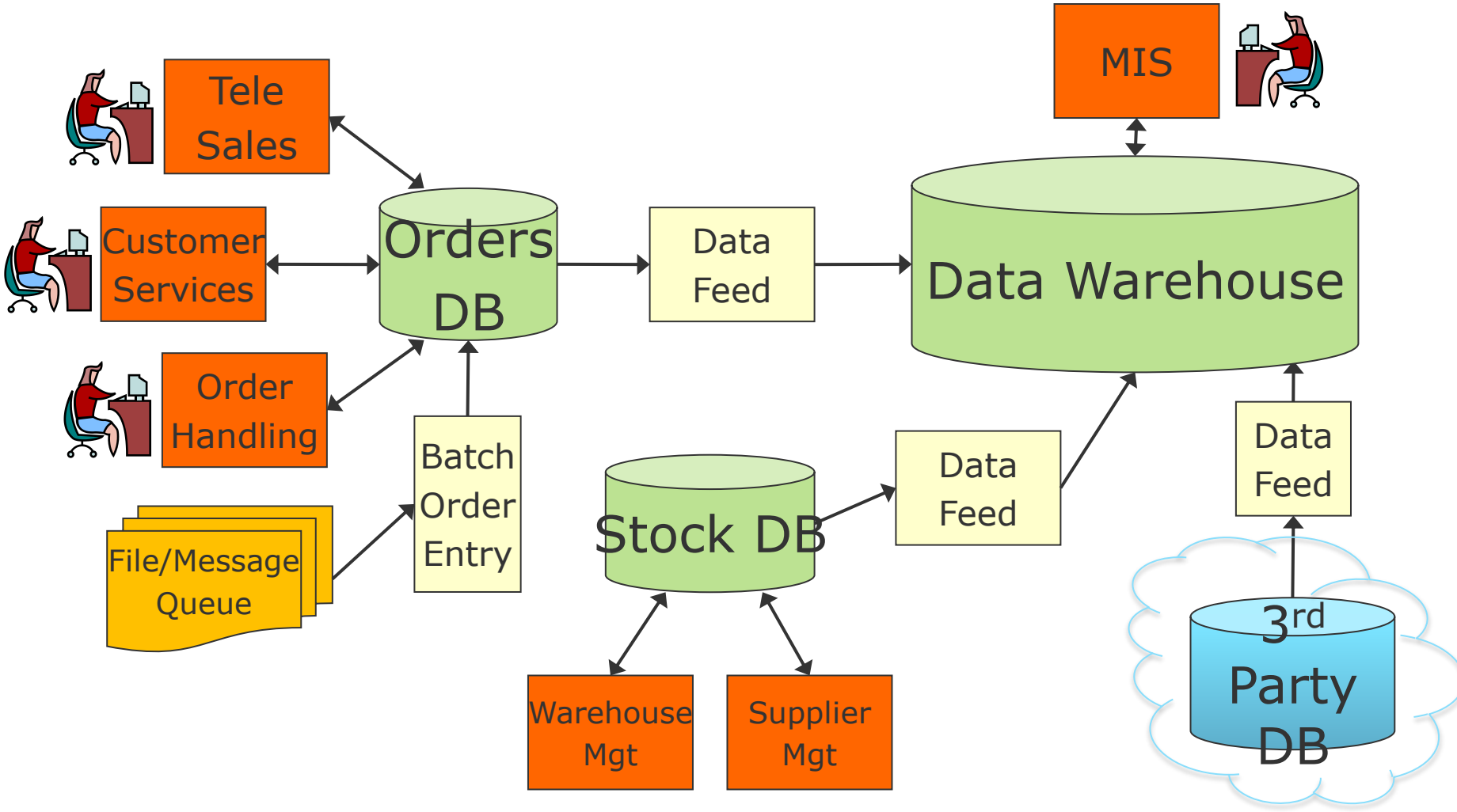
Evolution: New Application, Existing Data



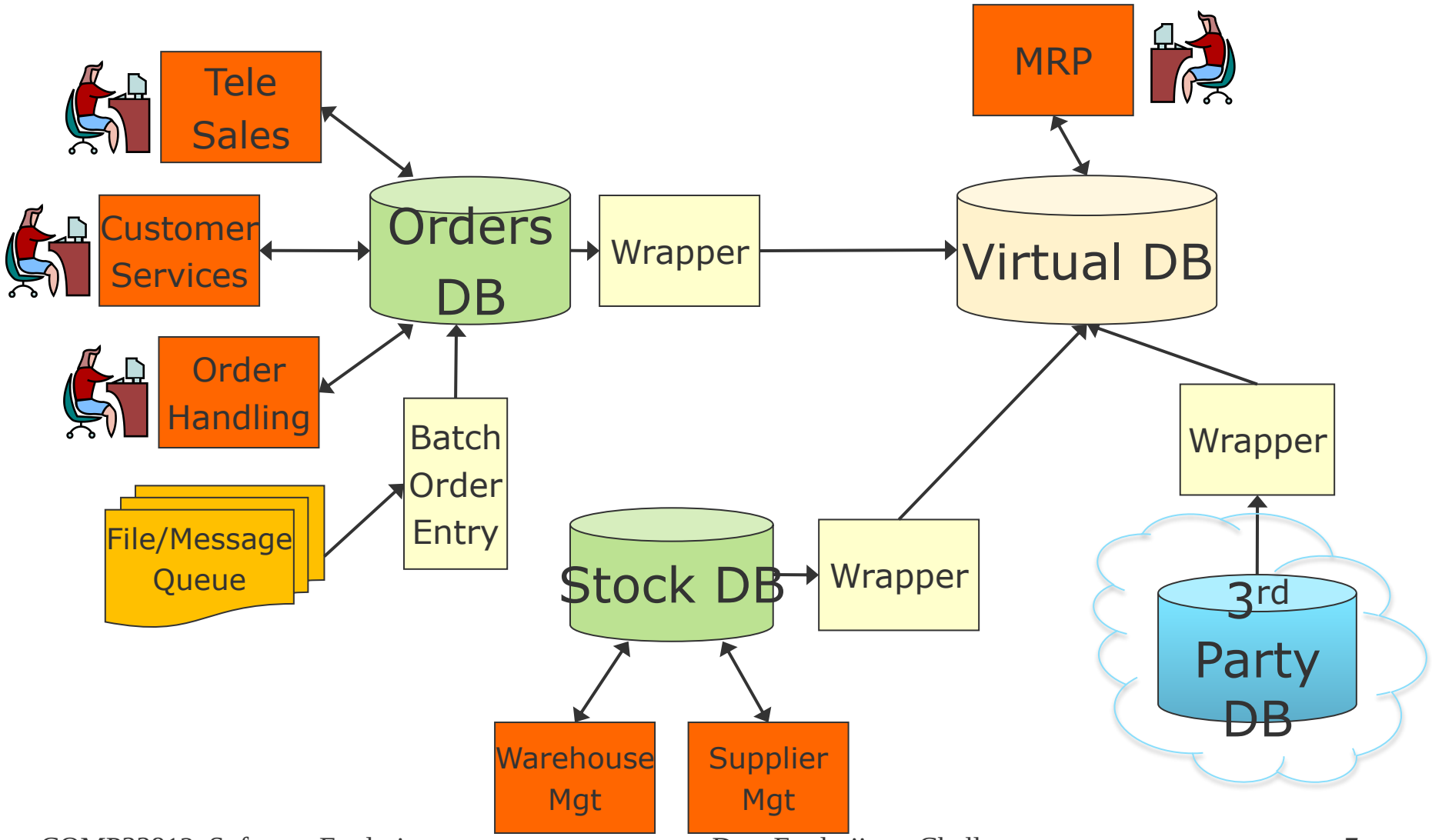
Evolution: New Application, New Data



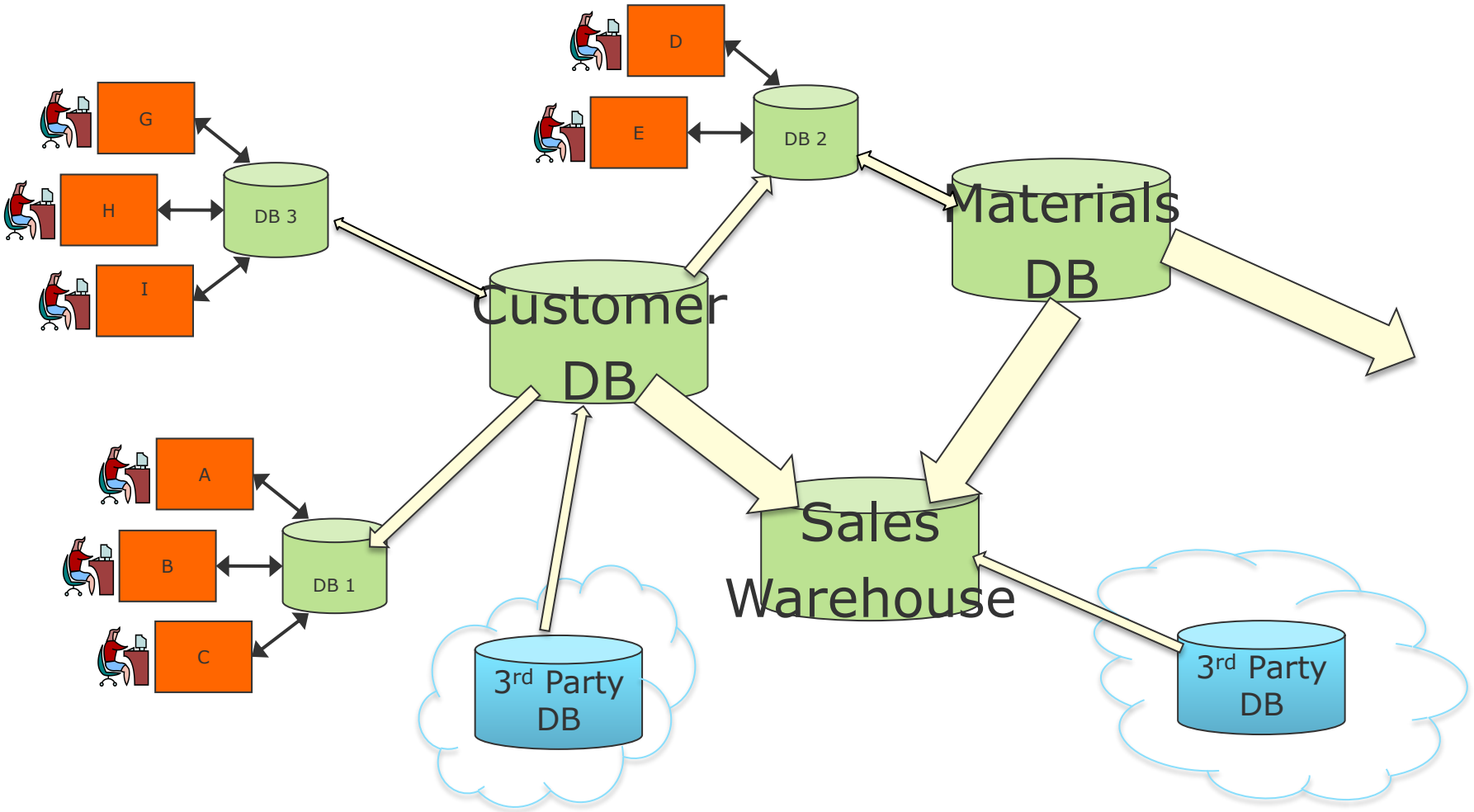
Evolution: Data Integration



Evolution: Virtual Data Integration



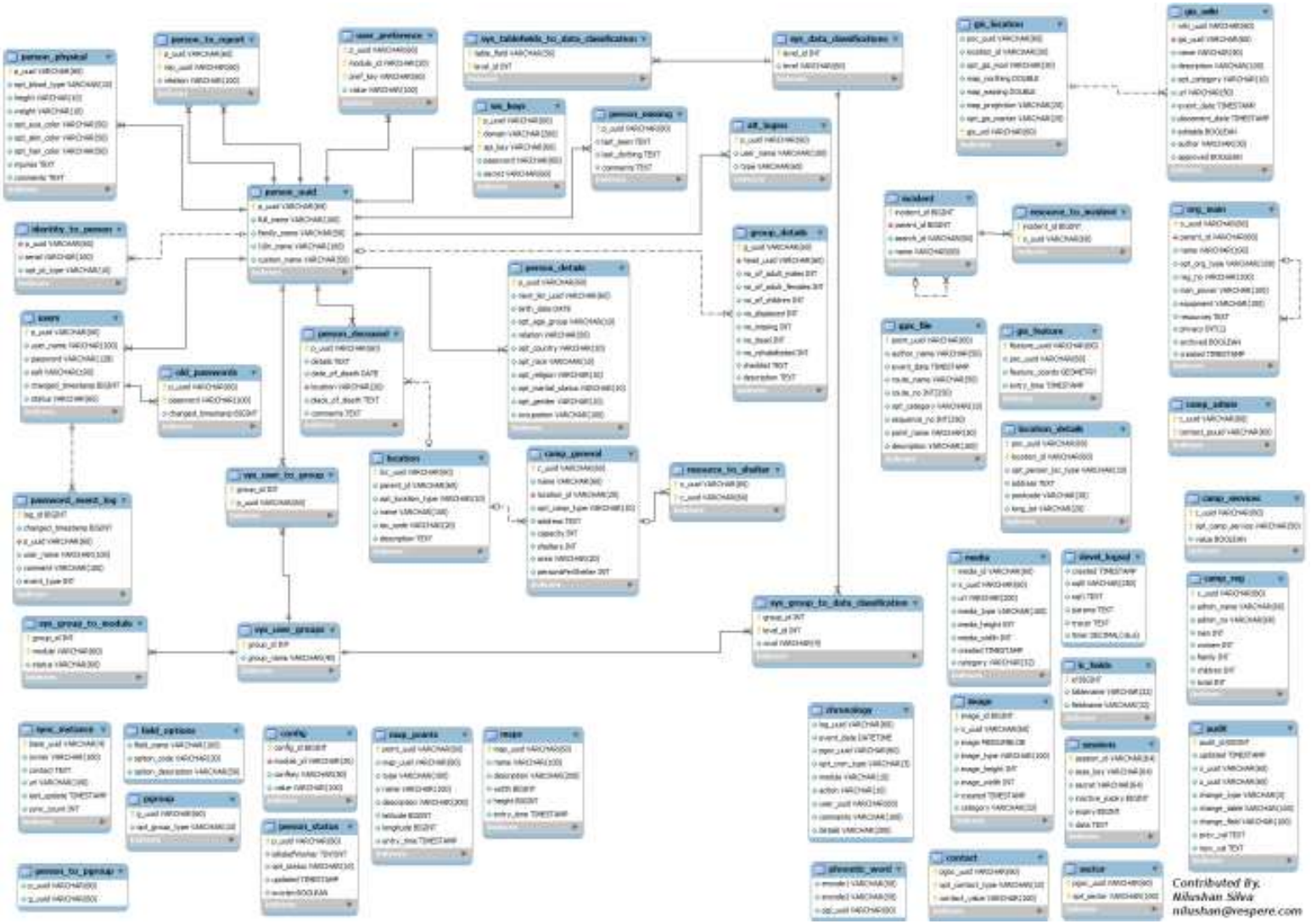
Evolution Patterns for Data



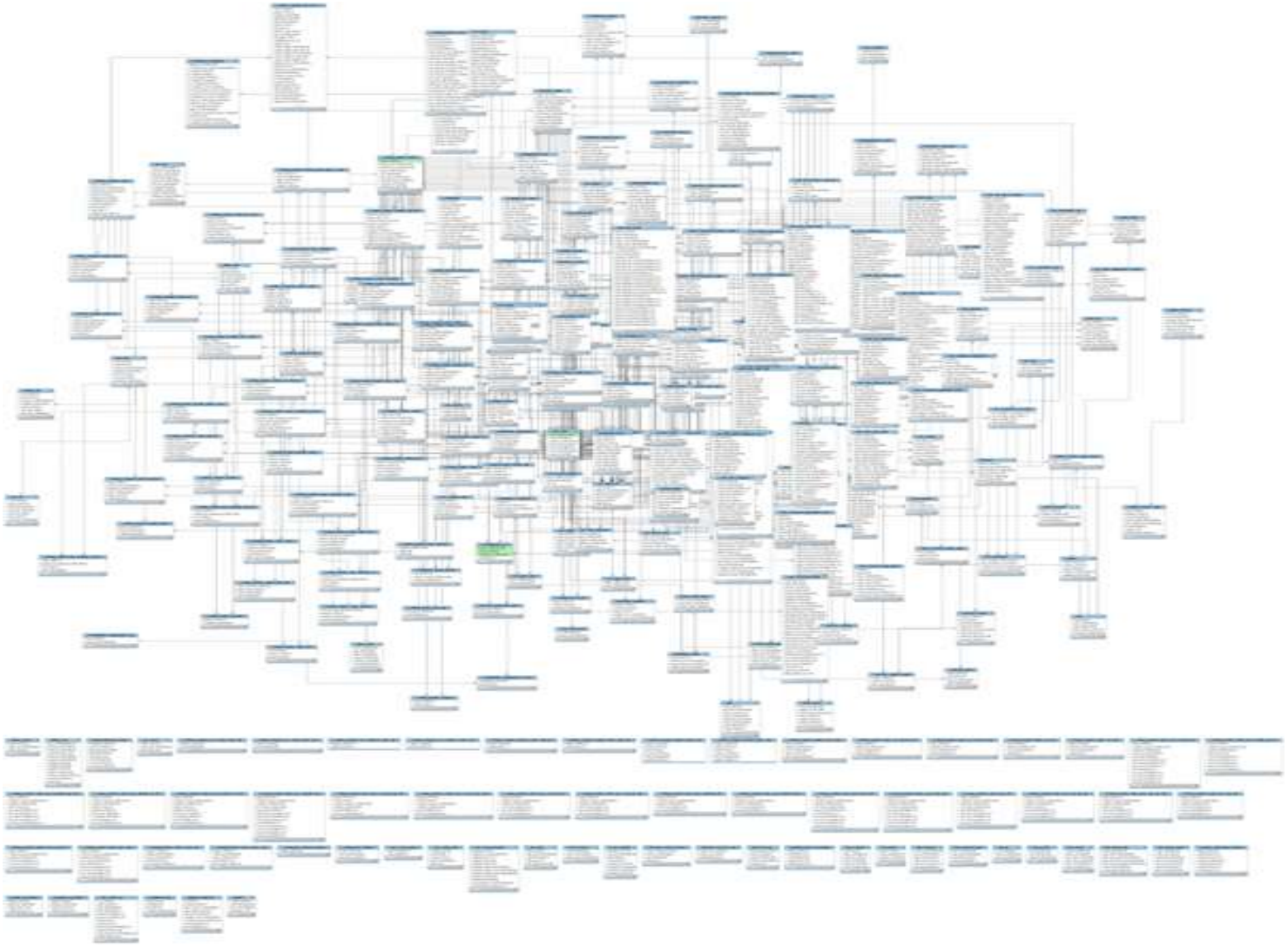
Evolution Challenges for Data-Intensive Systems

- Schemas are large, and often poorly structured
- High data volumes/velocity/variety
  - (Gartner Group's 3 V's)
- No one individual understands the whole schema/data
- Relationships between sources are complex
- Orders of magnitude more hidden dependencies than in software
- Data must record the historical picture
- Much of the data is mission-critical
- Much of the data is of very poor quality

Schemas: How Large is Large?

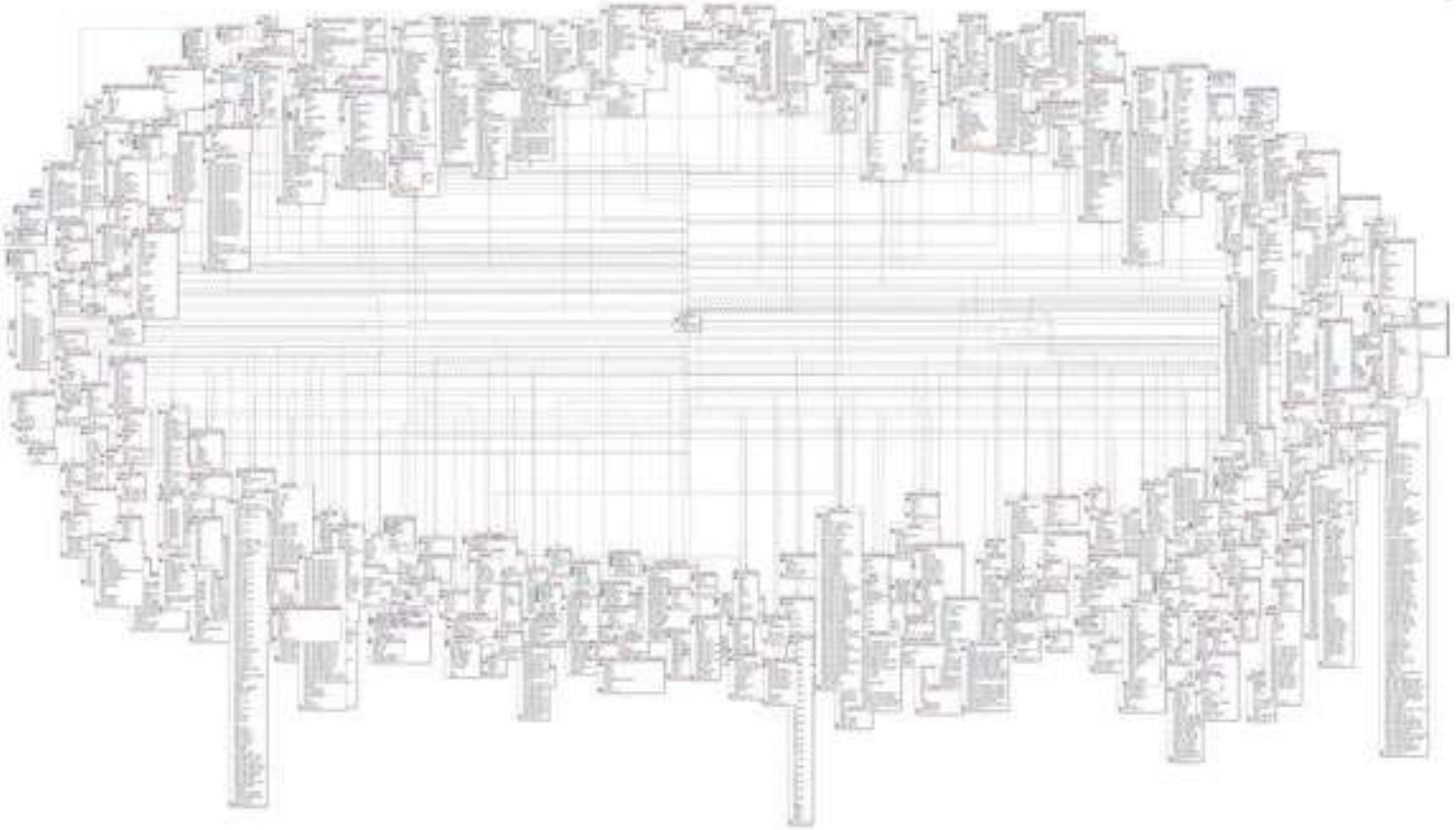


Understand?

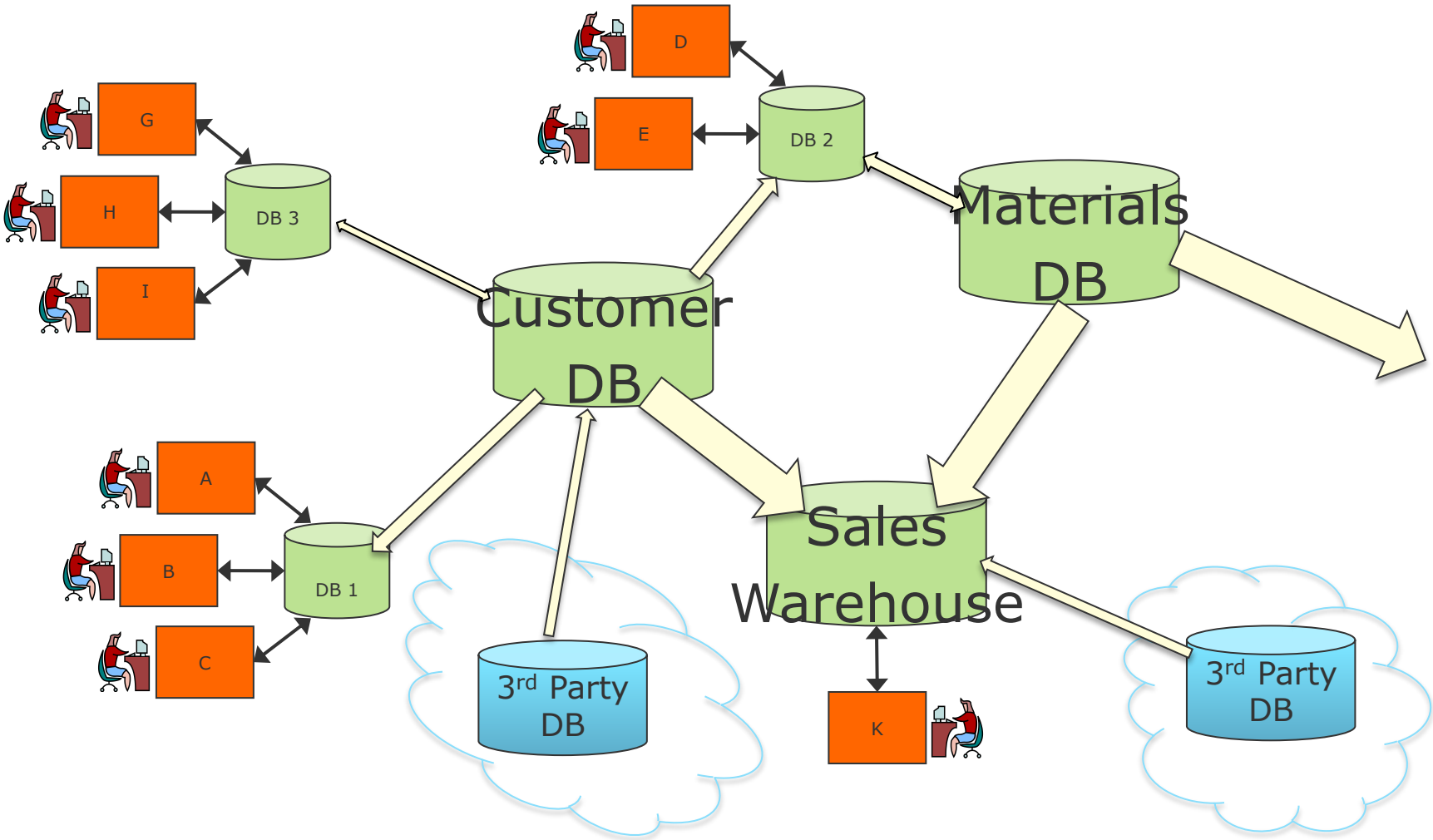


MANCHESTER 1824
The University of Manchester

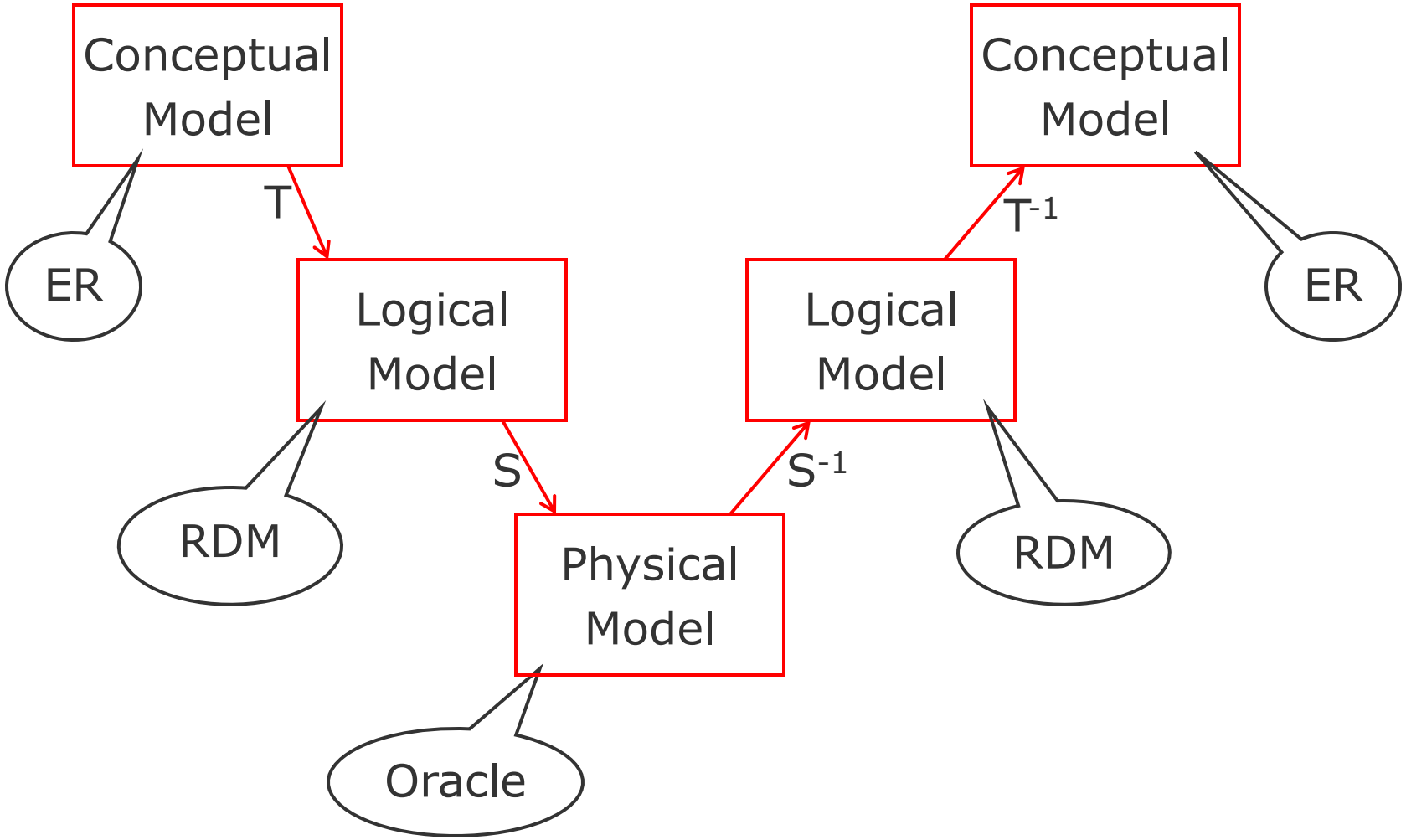
Understand?



Global Data Structure



Reverse Forward Engineering for Data



Must Preserve Historical States

- Software versions
  - Current version describes just the processes and business rules in force now
- Data entered 5 years ago still present in the system
  - Reflects business rules in force 5 years ago
- When business rules change, must implement them in a way that doesn't invalidate older data that doesn't fit them
  - Complicates integrity checking for the data
  - Complicates implementation of business rules
  - Complicates learning from the data (mining)

Poorly Designed Schemas

- Structure of data sources degrades over time
  - cf. software structure
  - Time pressures, Availability pressures

- As with software, many data sources are poorly designed in the first place
  - “Not only are many software engineers designing databases badly, but they are doing it in perversely creative ways.” Michael Blaha (2001)

- As with software, poor structure is a barrier to evolution and maintenance

Examples of Poor Structure (1)

- Failing to use surrogate keys
  - E.g. employee(Name, Address, NINum, Age)
- Representing Data as Metadata
  - E.g. representing multi-valued attributes as multiple fields

Student ID	Exam Mark	Coursework Mark
99123	51	13
99124	59	15

Student ID	Assessment Type	Mark
99123	Exam	51
99123	Coursework	13
99124	Exam	59
99124	LabMark	15

Examples of Poor Structure (2)

- Combining several types of data in one field

StudentID	Name	Address
99123	Fred Smith	24 Oak St, West Norton, Newburn, Borsetshire, U.K. NR3 5JG
99124	Julie Jones	104a Bat Lane, Chipping Oldhall, Borsetshire, OH3 6FE, U.K.

StudentID	Name	Addr_Line1	Addr_Line2	Addr_Line3	Addr_Line4	Addr_Line5
99123	Fred Smith	24 Oak St	West Norton	Newburn	Borsetshire	NR3 5JG
99124	Julie Jones	104a Bat Ln	Chipping Oldhall	Borsetshire	OH3 6FE	U.K.

Example of Limiting Structure

- Examine the following record structure

EmpID	Name	Dept	Area	Commission	Bonus
99123	Fred Smith	FinInd	Blessford	3.0	2,000
99124	Julie Jones	FinDom	Longston	4.5	1,579

- We now wish to pay commission on a sliding scale
  - e.g. 2% for first £100,000, 3% for next £100,000 and 5% for everything above that. Different departments will set different sliding scales for their staff

- How does the data structure have to change?

Volume of Data vs Size of Software

- Programs seem very small compared to the data stored in databases
  - Terabyte databases are common nowadays
  - Petabyte databases now exist
    - In 2010, data rate peaked at 10GB per second
    - 2008: CERN LHC will generate 15PB per year
    - 2015: CERN 75 petabytes from LHC in 3 years
    - 1PB = 250 bytes □ 1015 bytes
  - Exabyte databases are coming!

Consequences of Volume?

- RBS/NatWest/Ulster bank, June 2012
  - Every night, batch software applies transactions to accounts overnight.
  - More than 10 million accounts to update per night.
  - Around 20 million transactions per night.
  - Upgrade made to the scheduling software.
  - Problems surfaced during Tuesday night run.
  - Couldn't find log of what txs had been completed, so couldn't recover as normal.
  - Stopped the batch processing while they tried to find and fix the problem.

Consequences

- RBS/NatWest/Ulster Bank, June 2012
  - Problem finally fixed on Friday.
  - Had then to process all the transactions from Tuesday onward…
  - Effects on customers?
  - Cost for RBS?

Remedial Actions Taken by RBS

- £175 million set aside to cover costs and compensation associated with the failure.
- Extended branch opening hours:
  - Evenings until 7.00pm
  - Sunday
- Doubled the number of call centre staff
- Waived cash advance/one month interest for credit card holders
- Waived overdraft charges for affected customers

Lack of Knowledge

- Over time, organisations lose track of
  - What data they store
  - Where it is kept
  - Where it comes from
  - Who is responsible for it
- There can be a lot of duplication
  - I.e. both personnel and payroll record details of starting salaries
  - Data can easily become inconsistent
  - There can be a lot of wasted effort (£££)
- Very hard to understand full impact of change

Poor Quality of Data

- Much of the data stored in ISs today is:
  - Inaccurate
  - Inconsistent
  - Out of date
  - Incomplete
- Users continually complain that:
  - They cannot get the data they need
  - The data they need comes too late to be useful
  - They cannot use the data they get

Effects of Data Quality

- Data quality problems affect data architectures
  - corrupt query results
  - prevent successful integration
- Data migration strategies should account for these problems by adding appropriate quality control steps
- Can be hidden for many years