

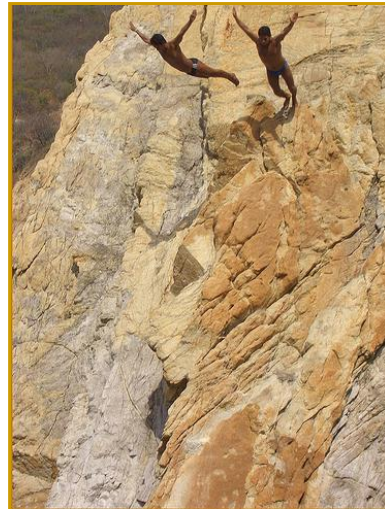
Enterprise Information Integration Technology: Architectures, Uses and Abuses

Mark Madsen
<http://ThirdNature.net>
mark@ThirdNature.net

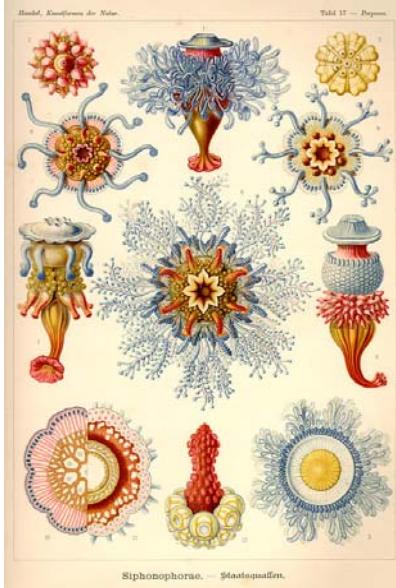
Intro & Outline



- Why we need new tools for on-demand integration
- What are the alternatives?
- EII definition and models
- How EII tools work
- Ways you can use and abuse the tools
- Some usage criteria



Problems Requiring New Approaches



The technical problems driving us to extend our integration tools:

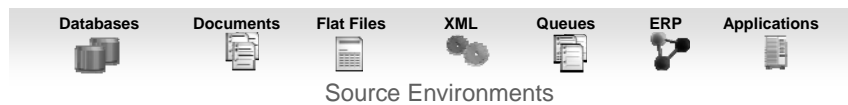
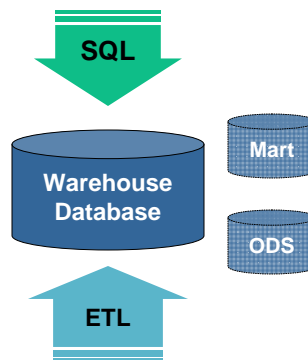
- Data currency
- Data source diversity
- Data type diversity
- Delivery via different APIs
- On-demand data consistency
- Regulation and data security

What We've Been Building

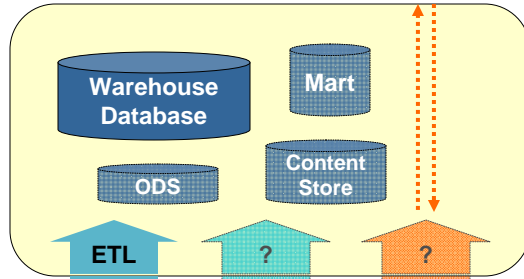


The original view:

- Data warehouse as a system or application.
- Cleansed historical data on a snapshot basis.
- Physical storage of all the data to be accessed.
- End user tools and infrastructure combined.

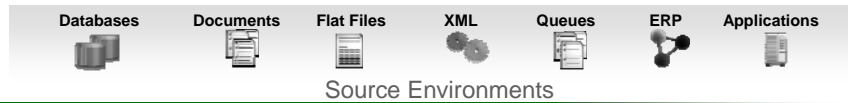


What We Need to Build Now

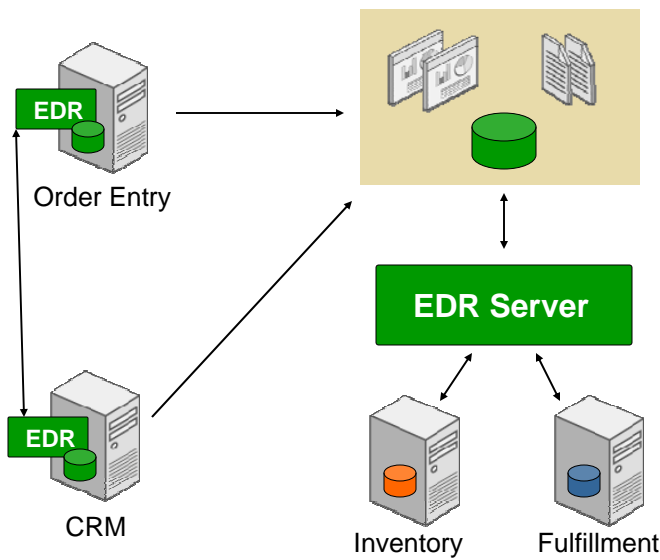


The new view:

- Data warehouse as a platform.
- Access to both historical and current data.
- Multiple storage methods, possibly distributed.
- Multiple access methods.
- Data usage decoupled from underlying platform.



EDR: Enterprise Data Replication



ETL: Extract, Transform and Load



The good points

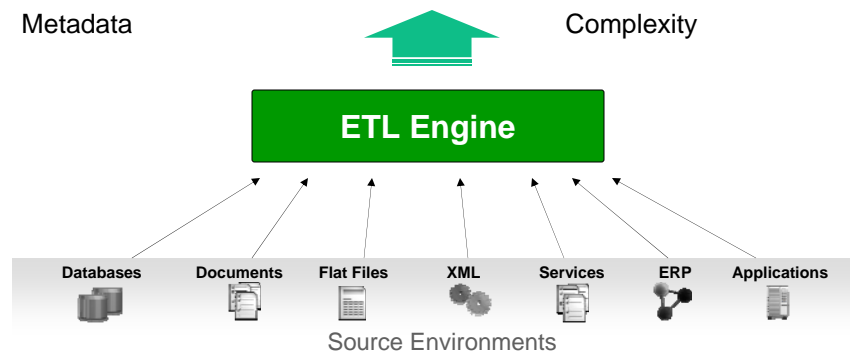
Connectivity
Transformation
Read-write access
Metadata



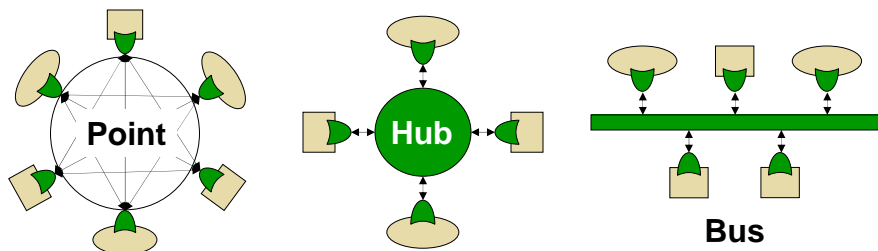
Database Targets

The bad points

Requires a physical target
Latency
Distributed query limitations
Complexity

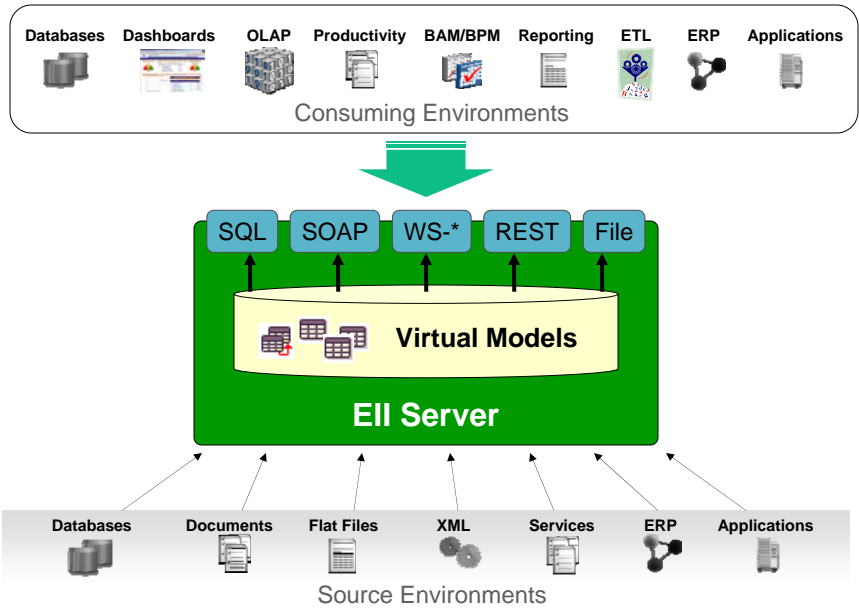


Enterprise Application Integration

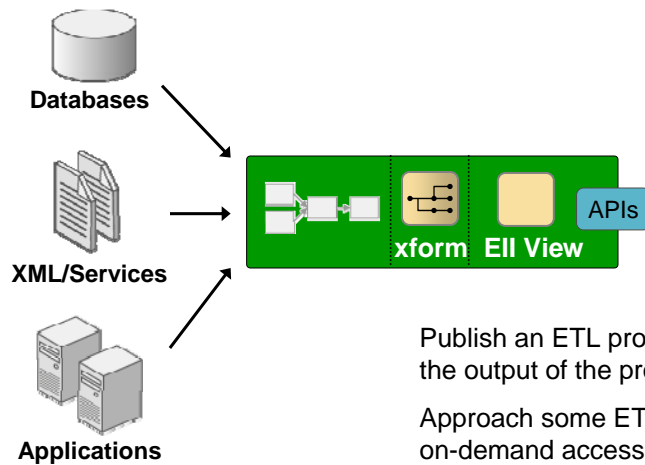


- Multiple models
- Multiple implementations
- Evolution toward open standards-based services and SOA

EII: Enterprise Information Integration



EII Architecture: Process-based

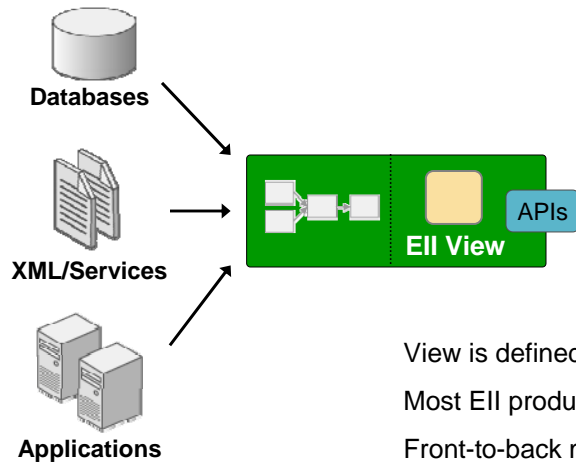


Publish an ETL process as a service – the output of the process is the “view”

Approach some ETL vendors have for on-demand access

Back-to-front model

EII Architecture: Model-based

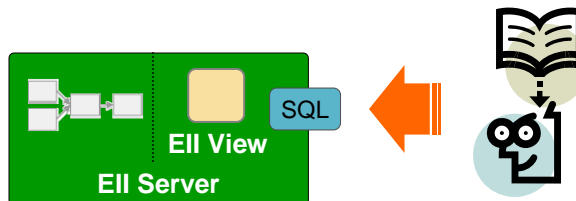


View is defined as a model over sources
Most EII products work this way
Front-to-back model

How EII Works



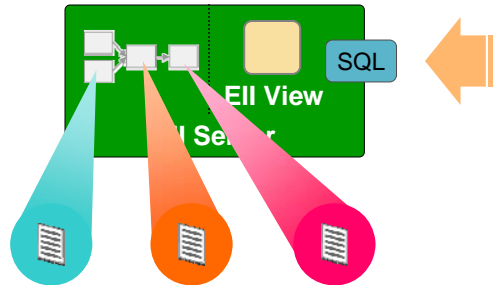
1. User's client sends a request which goes to the EII server interface



How EII Works



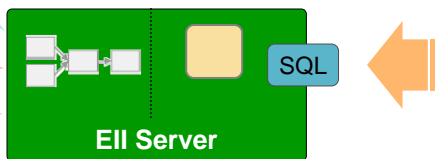
2. The EII server parses the request and creates query fragments to send to the sources



How EII Works



3. Each source system receives its query fragment, processes it and returns the result



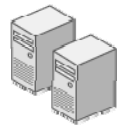
How EII Works



Databases

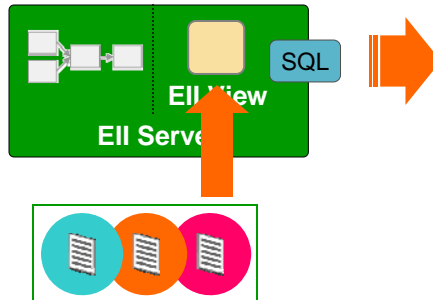


XML/Services



Applications

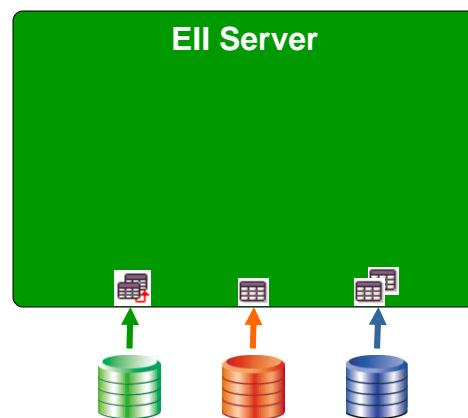
4. EII server assembles the final result (which may mean additional processing) and the result is sent back to the client



How You Work With EII Tools



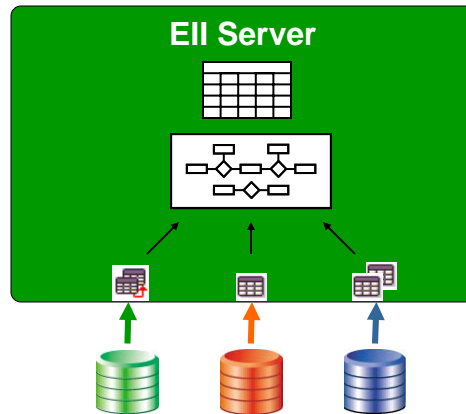
1. Define data sources
2. Define integrated view
3. Map source data into the integrated view
4. Publish view
5. Tune query*
6. Query via the published interfaces



How You Work With EII Tools



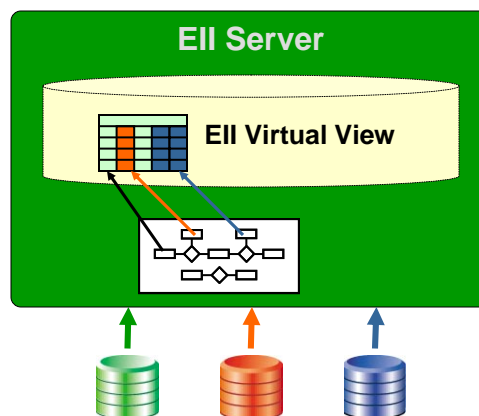
1. Define data sources
2. Define integrated view
3. Map source data into the integrated view
4. Publish view
5. Tune query*
6. Query via the published interfaces



How You Work With EII Tools



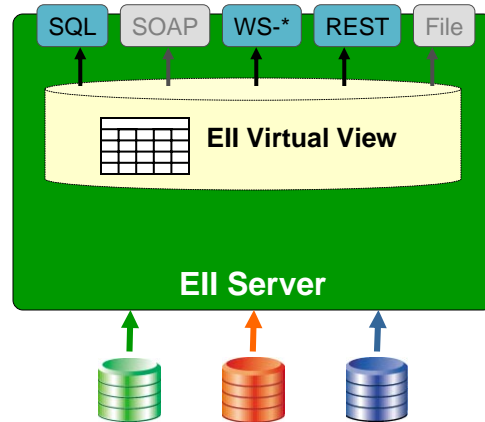
1. Define data sources
2. Define integrated view
3. Map source data into the integrated view
4. Publish view
5. Tune query*
6. Query via the published interfaces



How You Work With EII Tools



1. Define data sources
2. Define integrated view
3. Map source data into the integrated view
4. Publish view
5. Query via the published interfaces
6. Tune query*



What Sorts of Problems Can We Solve?

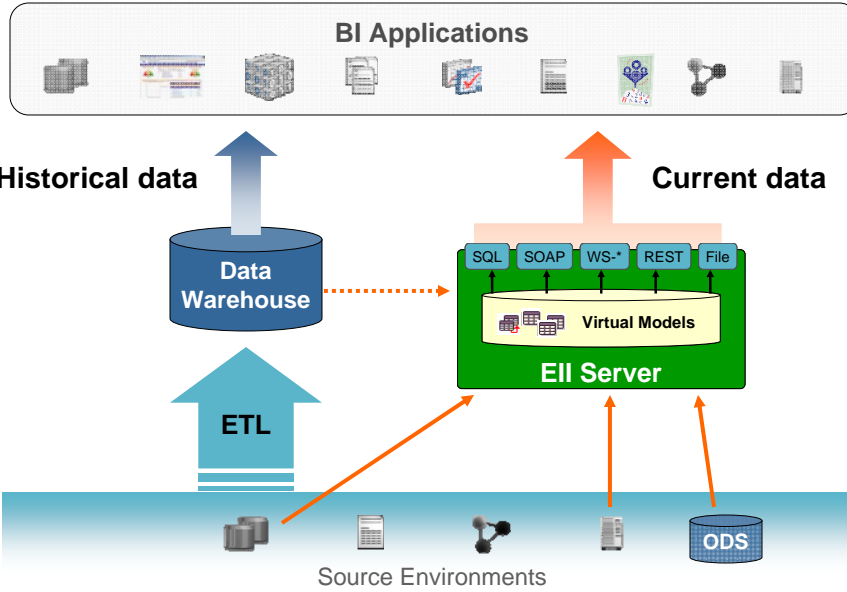


Applications where you are being asked to provide any of the following:

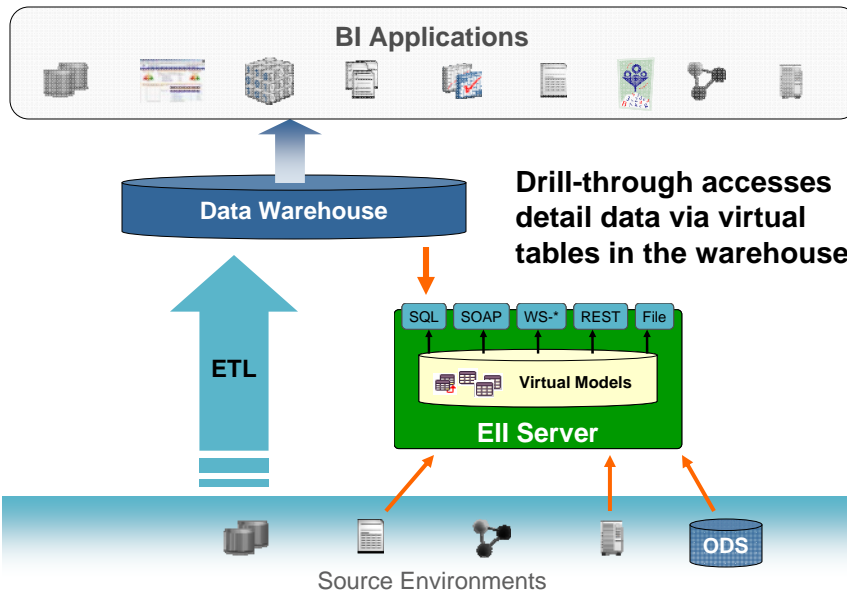
- More up-to-date data
- Data in alternate formats
- Data in non-relational formats
- Data for portals, web sites or composite applications



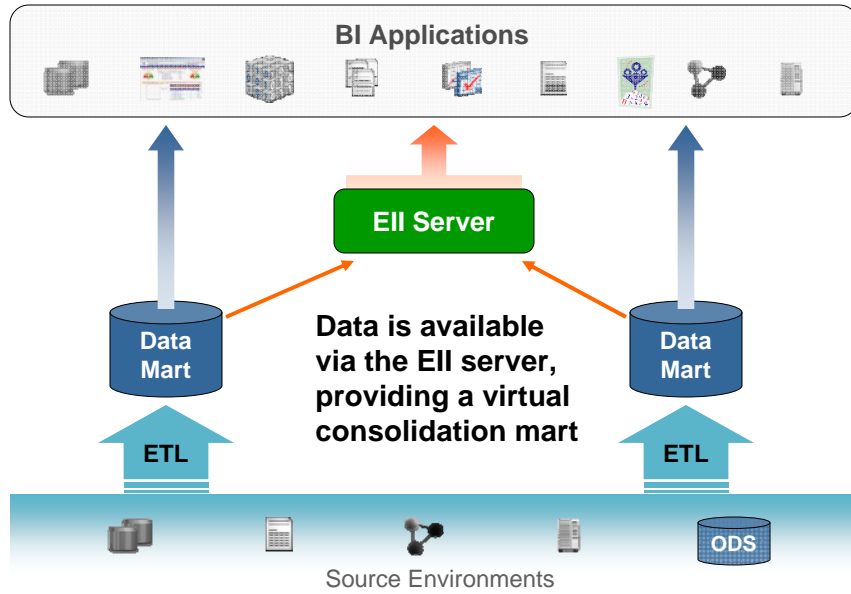
Adding On-demand BI Capabilities



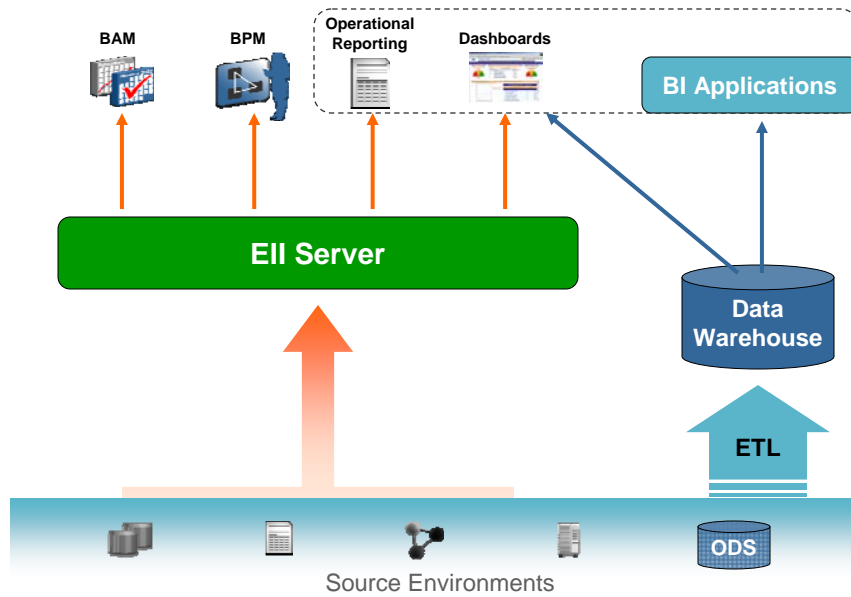
Drill-through to Source Detail



Consolidated Views



Performance Management and BI



Compliance and Regulatory Reporting



- Many compliance and reporting solutions deal with process, content and tracking documents.
- Typical characteristics:
 - Historical data
 - Non-BI data
 - Multiple source applications
 - Documents and content
 - Needs vary over time

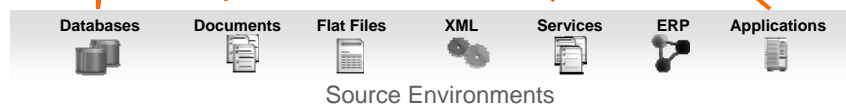


Portals, Composite Applications and SOA



Typical Portal scenario:

- Disparate data
- Heterogeneous sources
- Point integration
- Minimal reuse



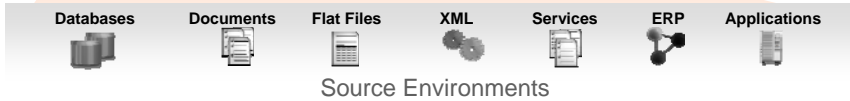
Portals, Composite Applications and SOA



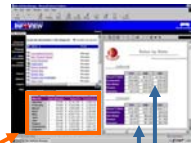
EII provides a “data services” layer

- Standards-based interfaces
- Single view of source data
- Single point of integration
- Reuse of data as a service

Data Services (EII)



Portals, Composite Applications and SOA



Data Services (EII)

Transaction Services (EAI)



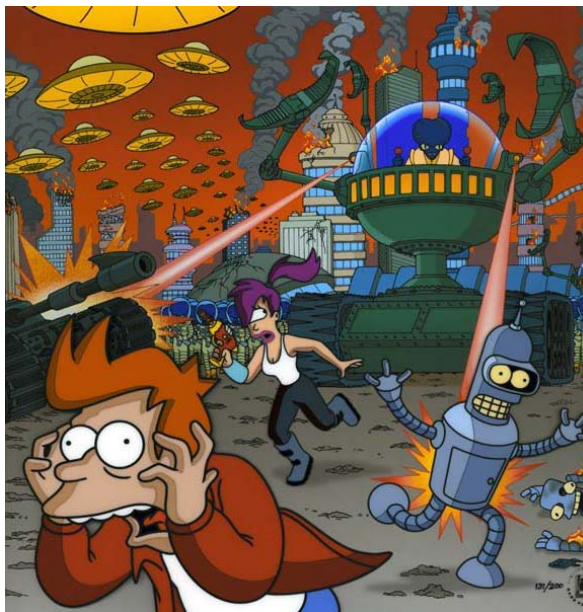
Rapid Prototyping



- If you begin building a data services layer, it can make the prototyping of new data-intensive applications go much faster.
 - No need to physically model, extract and store data for a prototype application.
 - Simple to set up
- Examples:
 - Adding a subject area
 - Adding a star schema
 - Adding data to a portal



Just Remember It's a Prototype



Going Further: the Virtual Data Warehouse



TDWI EI Series, Feb 2007

Copyright © 2007 Mark Madsen

Slide 31

Other Questionable Ideas



- Getting around pesky rules and procedures:
 - Like being direct query access cutoffs
 - Or security
 - Or development controls regarding data access
- You can even treat an EII view like a data source for copying data...



TDWI EI Series, Feb 2007

Copyright © 2007 Mark Madsen

Slide 32

EII Isn't a Miracle Cure



- It can help you take shortcuts you shouldn't.
- Some other shortcomings you may face:
 - Metadata sharing and interoperability
 - Reliability (failure modes, not the tools)
 - Performance considerations



Some Guidance: When Does EII Fit?



There are some basic criteria and tradeoffs to consider:

- Data currency vs. latency
- Diversity of data sources
- Data cleansing & transformation
- Predictability of performance
- Access to the same data is needed via different interfaces
- Non-relational sources
- Frequency of access
- Data volumes
- And more...