# Two Guys and a Transformation

## ETL Best Practices and Techniques

Gabe Villa, Director of Consulting, Datalere
Marc Beacom, Managing Partner, Datalere

# Marc Beacom

## Managing Partner, Datalere

/marcbeacom @marcbeacom https://Datalere.com

## Experience

10 years DW/BI Consultant

20 Years overall experience

## Current Projects

Data Warehouse

Data Integration

Data Science

## Interests

User Groups

SQL Saturdays

PASS Summit

# Gabe Villa

## Director of Consulting, Datalere

/gabevilla @extofer https://Datalere.com

## Cloud and Data Architect

Skilled leader, architect, and technical expert focusing primarily on Microsoft technologies and is passionate about open-source technologies for integration, automation, and development.

## Colorado Transplant

Originally from El Paso, Texas, Gabriel lives in Colorado Springs with his wife and kids, where he is a Director of Consulting for Datalere, LLC., volunteers in the tech community and enjoys the Colorado outdoors.

## Microsoft Certified Professional

- MCPD, ASP.Net Developer
- MCTS, SQL Server Database Development

# Poll – Are you an…

- ETL Engineer ?
- DBA ?
- Other ?

# #01: ETL Templates

- Start developing right away with a known and consistent framework
- Pre-built with auditing/logging – just copy and paste to reuse the template
- Logging
  - Batch/Package/Table, Start/End dates for durations, Load Windows
- Auditing
  - Row counts, log rows with batch ID, rollback if needed
- Template Examples
  - Master / Parent Package, Child, Loading flat files

# ETL Templates

## Demo

# #02 - Standards

- Leverage system variables where possible such as logging
- Assists in troubleshooting
- Create and Follow Naming Standards
- Checklists for
  - Code reviews
  - Environment setup – servers and development stations

# #02 -Standards: Checklists

### Review Checklist

The following check list is used when reviewing an ETL package prior to promoting it from Dev to QA.

#### *Control Flow*

1. The major and/or minor versions have been adjusted accordingly
2. Precedence Constraint exists after the “SQL Load Check Status” task
3. All Control Flow tasks are enabled
4. All Control Flow tasks have the proper prefix
5. Project configurations being used

#### *Data Flow*

6. All Data Flow transformations have the proper prefix
7. Data Flow matches a documented Data Flow Pattern
    a. If the Data Flow does not match a Data Flow Pattern, an annotation should be added to explain why and what the unique reason is
8. Data flow does NOT contain blocking transformations – Unless a valid reason and donuts are donated to the team prior to code review
9. Lookup transformations are set to partial cache unless no cache is needed.

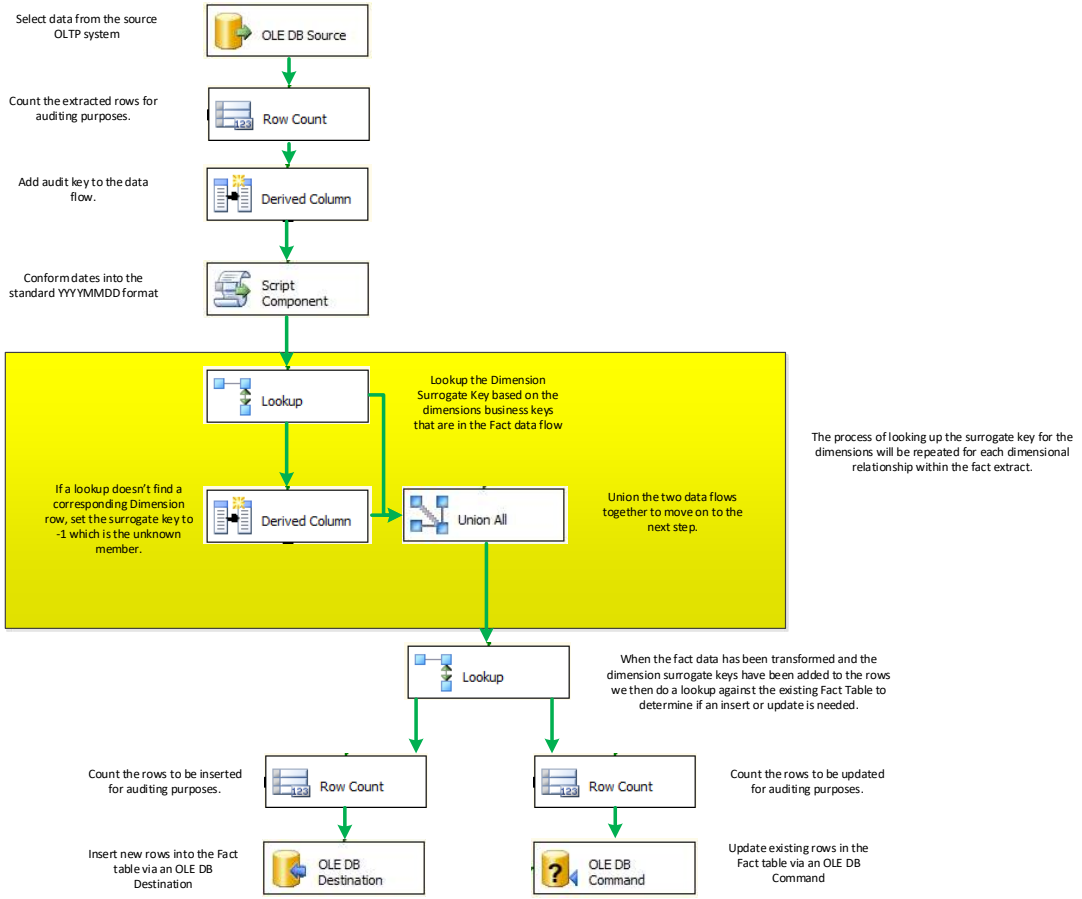
#### *Event Handlers*

10. The SQL Log Error task exists at the Package level for the OnError Event Handler.
11. All tasks in the Event Handlers are enabled and fully functional.

# #03 - ETL Load Patterns

- Your ‘Blueprint’ for ETL development
- Ensures consistency when developing packages
- Reduces long-term maintenance costs
- 3-4 ETL patterns for a typical Dimensional Model
- Socialize patterns with all ETL developers

# #03 - ETL Load Pattern Example



# ETL Load Patterns

## Demo

# #04 - Documentation

- ETL, while visual, isn't self-documenting
- High level and not line item documentation
- Gets new team members up to speed quicker
- Include patterns, templates, standards, checklists, etc.

# #04 - Documentation

**Table of Contents**

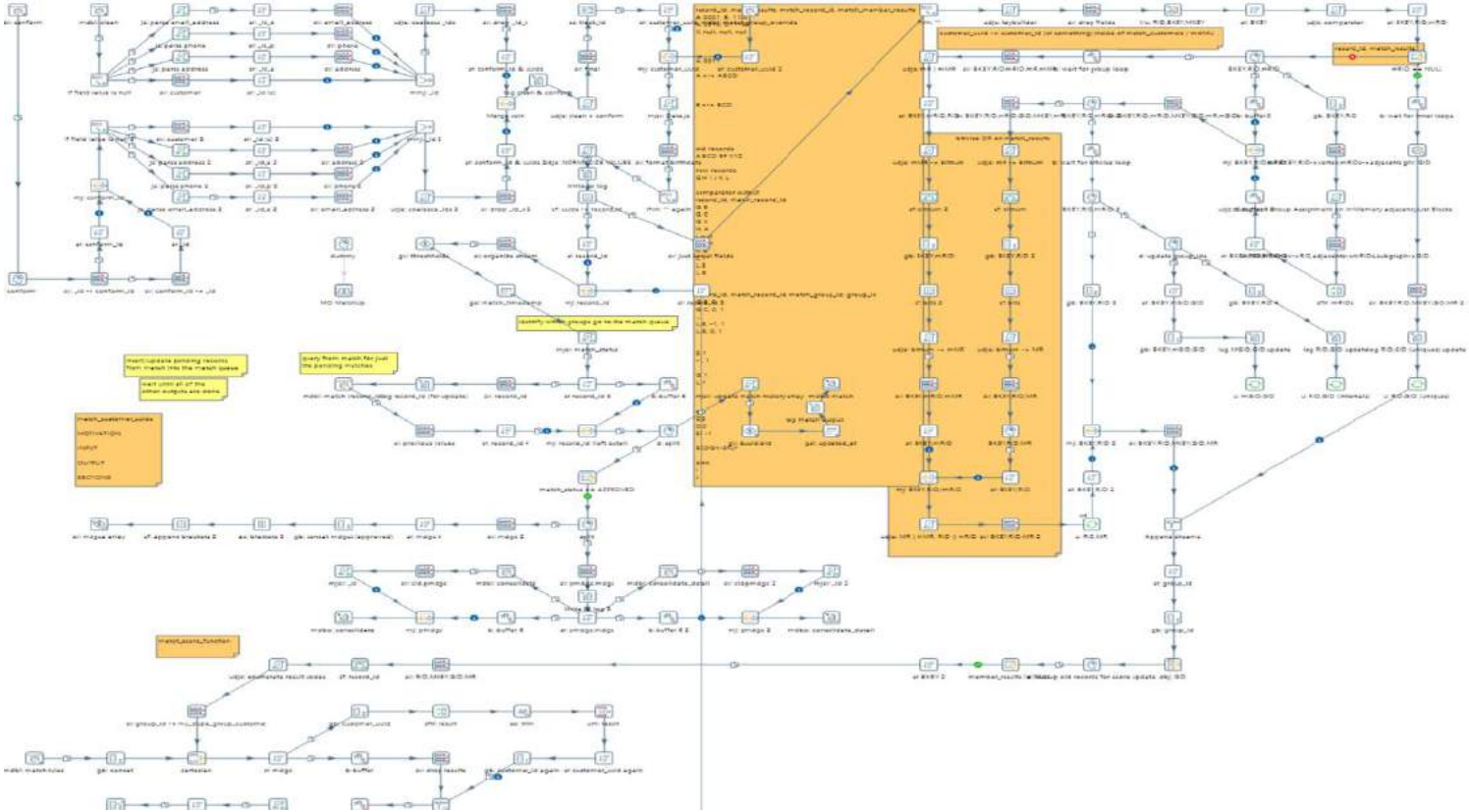
# #05 - Address Bad Data

- What is bad data? Who should define this? = You and Business!
- Develop a process, either manual or automated, to address bad data
- The outcome should be standardized and documented
- Options
  - Ignore or discard
  - Insert and flag
  - Redirect to another table/object

# #06 - Decouple Logic

- Reduce code down to small segments of functionality
- Easier to test and validate that you have met the finish line
- Reuse the code rather than copying and pasting the same code
- Don’t go overboard

# #06 - Decouple Logic



# #06 – Decouple & Performance Case Study

## Challenges

- Education Data Management ISV could not successfully match and load data
- Data matching and loading often took a day to process
- Need to refactor and optimize current ETL

## Recommendation

- Defined a load pattern that simplified the overall process
- Data cache adjustments allowing quicker lookups
- Sorting and Joining on database

# #07 - Format and Organize

- Keep things simple but more may be less
- Add comments / annotations where needed
- Follow team standards

# Formatting

## Demo

# #08 - Incremental Loading

- Reduces bandwidth and times during loads – data sizes are growing!
- Achieve near real-time data refreshes – up to 2 minutes
- Options
  - CDC
  - Last modified dates – beware!

# #09 - Parallel Processing

- Take advantage of idle resources
- Control flow
- MaxConncurrentExecutables property
- Default of -1 = Processor count + 2
- My Default = Processor count - 2
- Data flow
- EngineThreads property
- Default of 10
- Don’t overload the server resources

# #09 - Parallel Processing Case Study

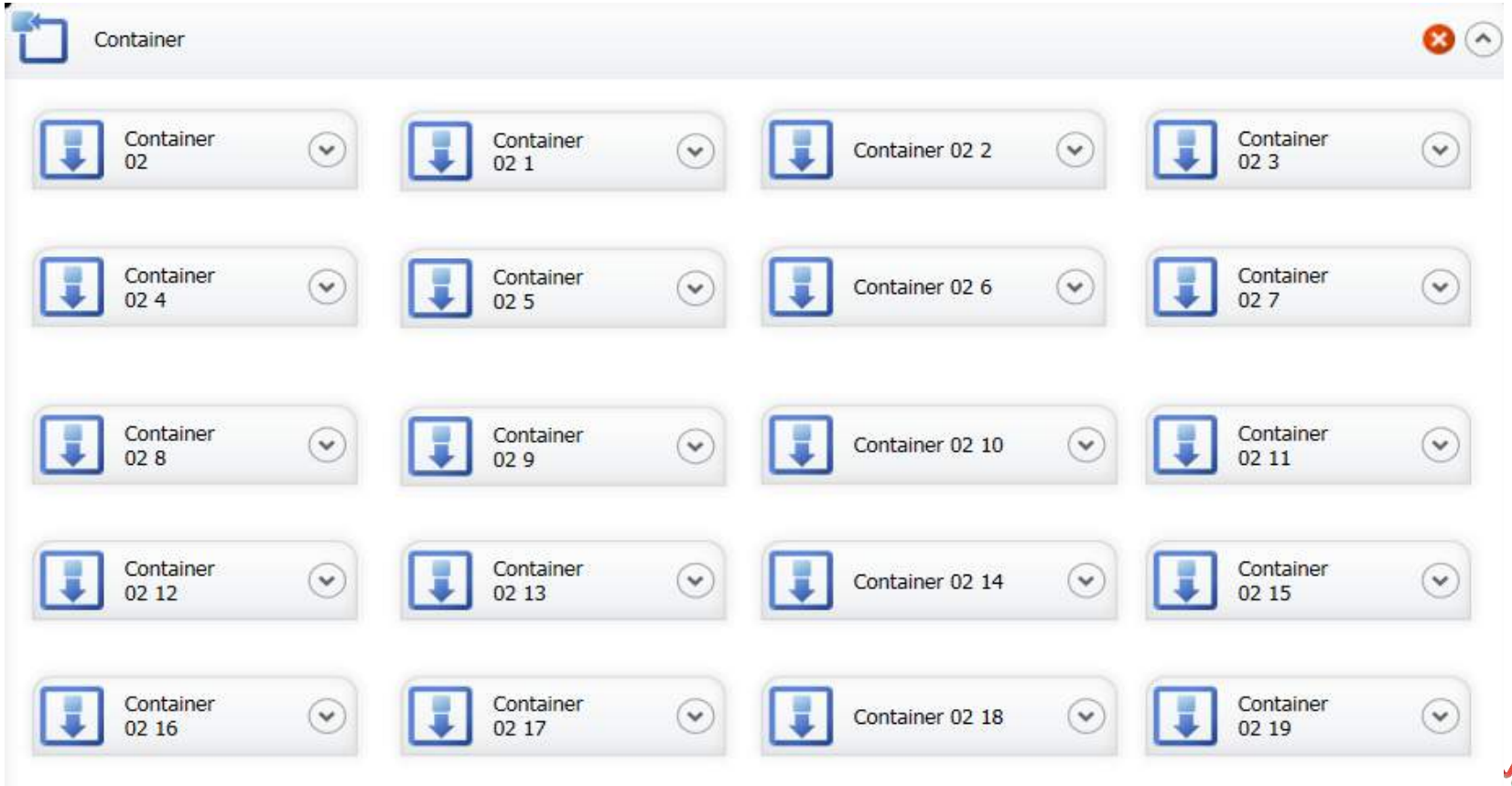
## Challenges

- 1 Billion daily rows
- Current load was 6.5 hours but had 3 Hour load window
- Consumed a large percentage of resources while processing

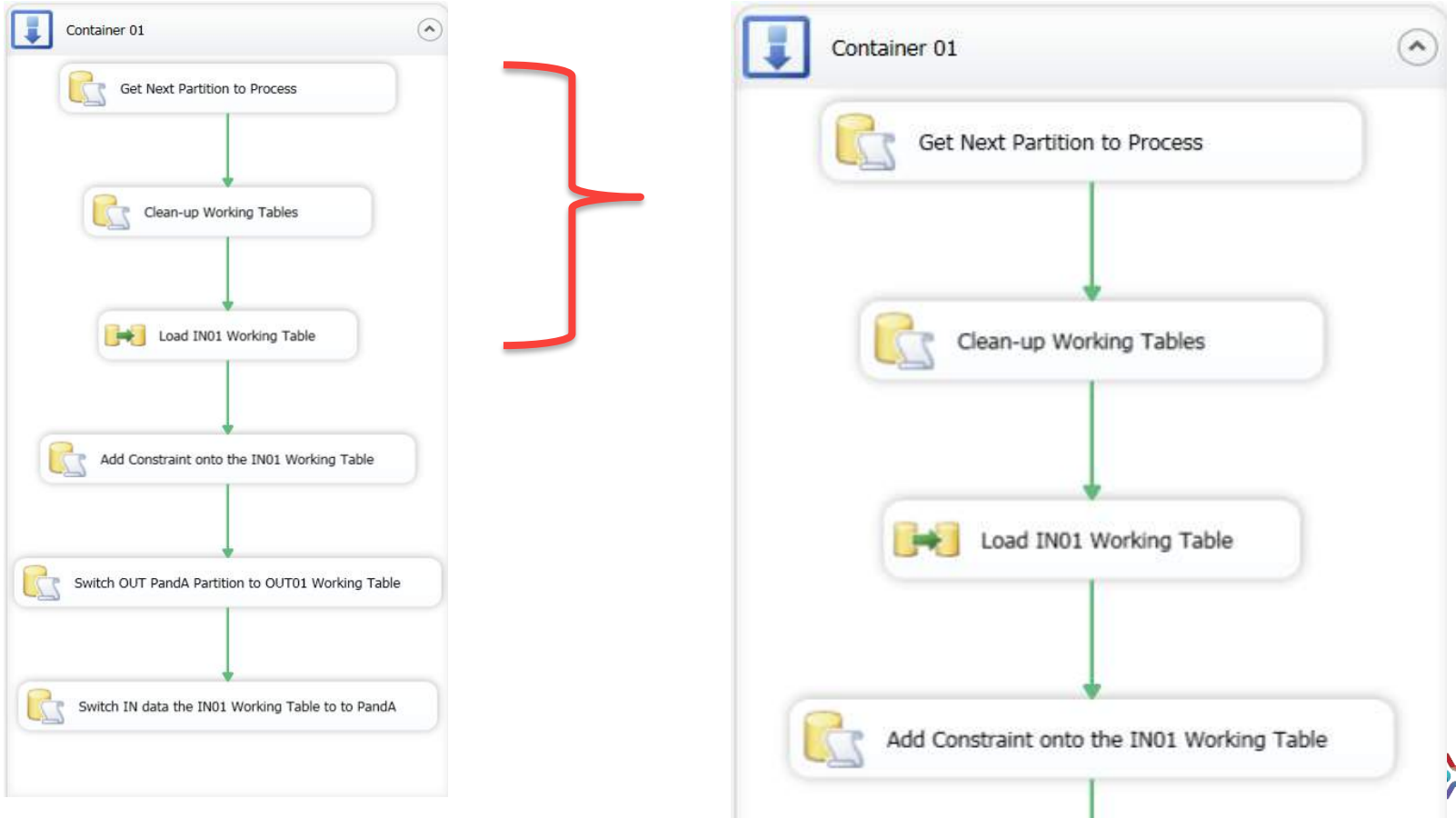
## Recommendation

- Partitioned table
- Parallel processing – Control flow
- Better data types – GUID to BigInt – Saved 12 GB / column
- Page level compression

# #09 - Parallel Processing



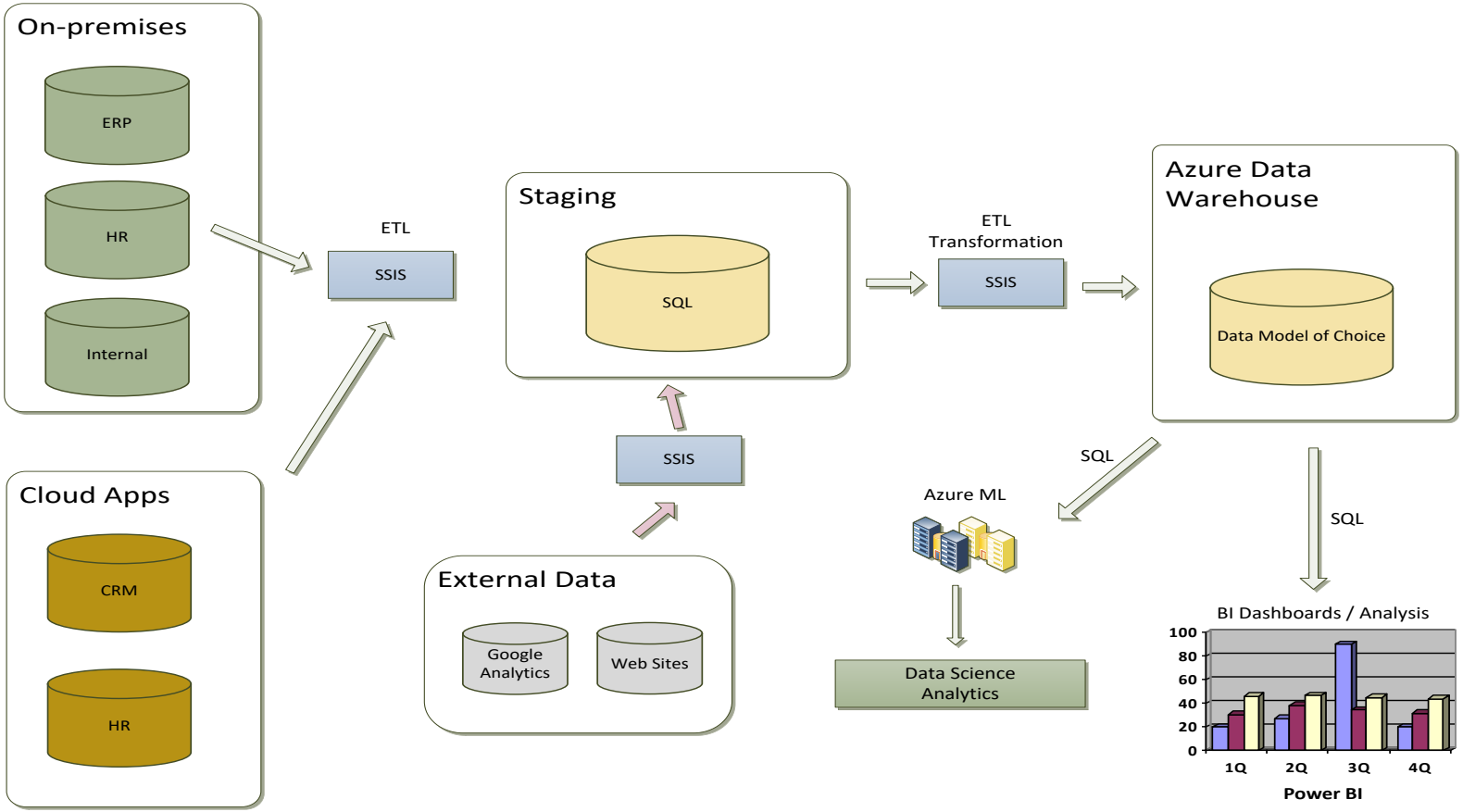
# #09 - Parallel Processing



# #10 - Stage data

- Reduces overall ETL complexity
- Little to no transformations out of source
- Reduce the ‘hit’ on the source system
- Separate database where possible and different schema if not

# #10 - Stage data



# #11 - Optimize Source and Destinations

- Only pull in columns you need
- Only pull in data rows you need – add a where clause
- Don’t use the select from table or view, always specify columns
- Index optimization – don’t forget about lookup queries
- Distributing data across multiple source flat files
- Sorting & Join in SQL – Sort Transformation is blocking!

# #12 - Blocking Transformations

## Blocking

- Fuzzy Grouping/Lookup
- Aggregate
- Sort


## Partially Blocking

- Merge Join
- Union All
- Lookup


## Non-Blocking

- Derived Column
- Data Conversion
- Row Count

# Blocking

## Demo

# #13 – Validation Framework

- Test data to ensure accuracy
- Developed my first validation framework in 2009 with SSIS
- Start small and build upon it
- Row counts
- Aggregations
- Compare known data with Data Warehouse data
- Compare source and DW data
- Can add a significate cost in development

# #14 – Bonus

- Know your requirements and where the finish line is
- Build a development plan which includes testing!
- Source Control check-in often and at least daily

# Thank You

## Learn more from Gabe and Marc


@ extofer

gvilla@Datalere.com

@marcbeacom

mbeacom@Datalere.com


PASS