# FROM BATCHES TO STREAMS WITH APACHE KAFKA

Allen Underwood – CodingBlocks.NET – @CodingBlocks – @theallenu

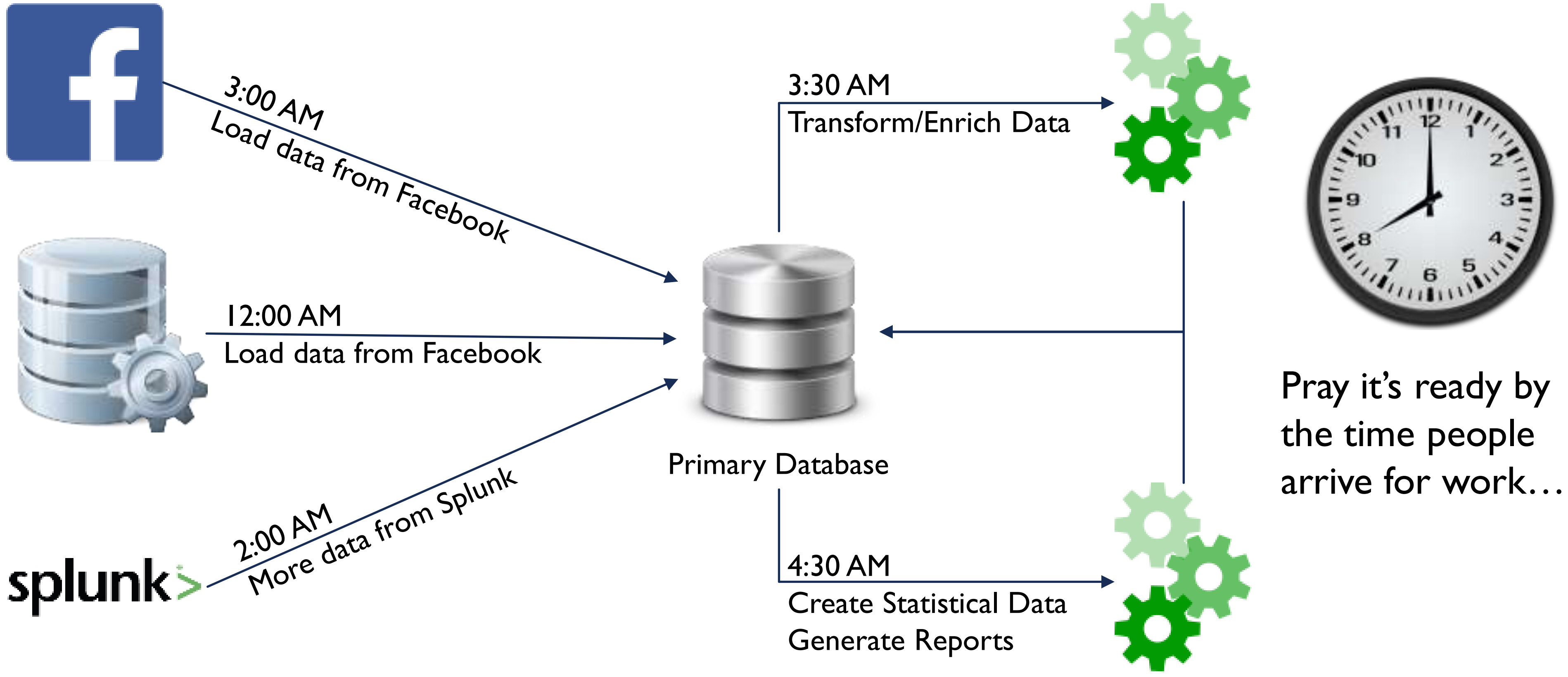
# DONUTS…

OR IS THAT DOUGHNUTS?

# DIY DONUTS

# STREAMING DONUTS…

https://youtu.be/SXEsDq7JAMI?t=109


3:00 AM
Load data from Facebook

12:00 AM
Load data from Facebook

2:00 AM
More data from Splunk

Primary Database

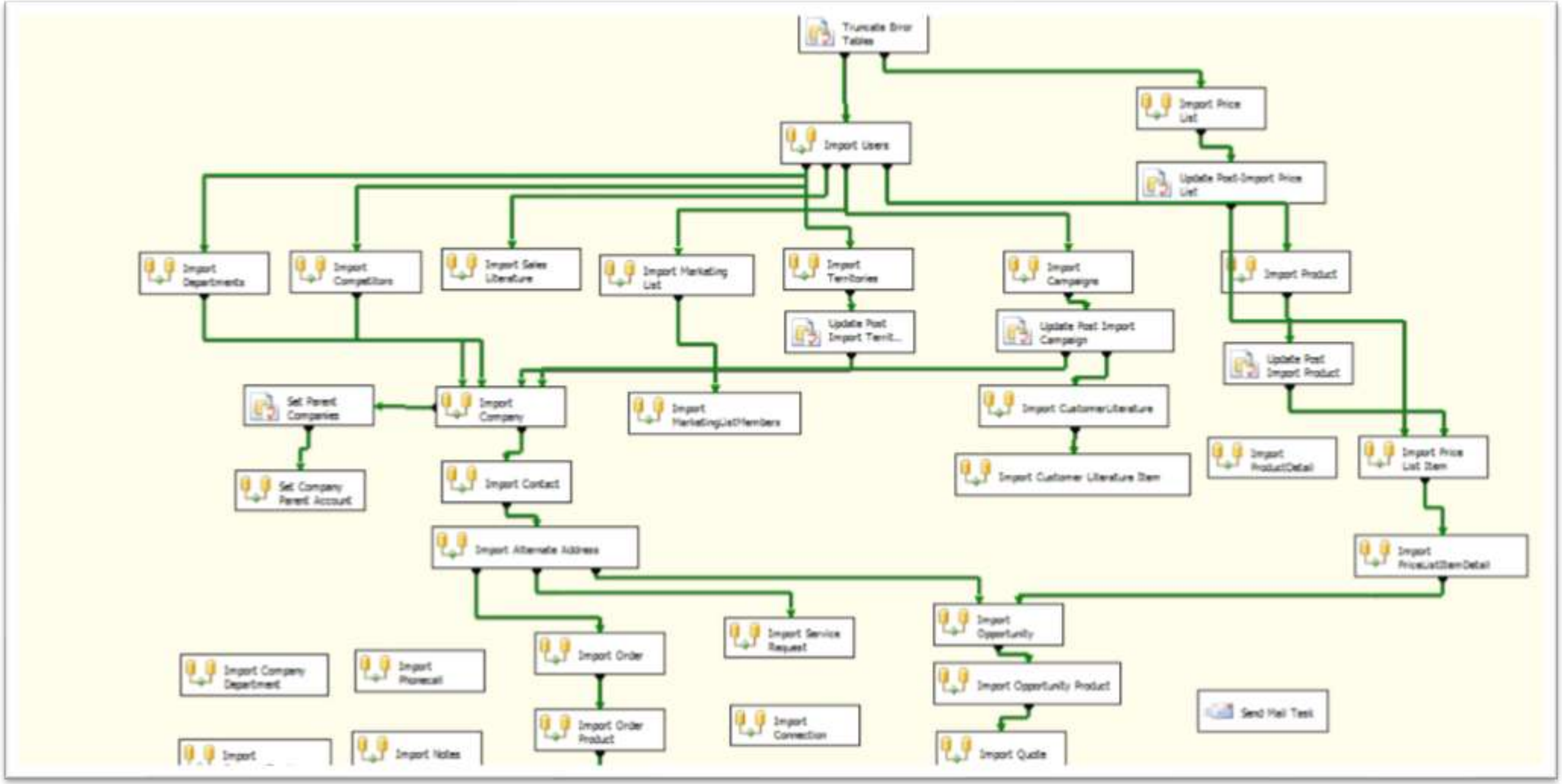
3:30 AM
Transform/Enrich Data

4:30 AM
Create Statistical Data
Generate Reports

Pray it's ready by
the time people
arrive for work…

# STANDARD BATCH ETL PROCESSES

# A NOT UNCOMMON SSIS PACKAGE

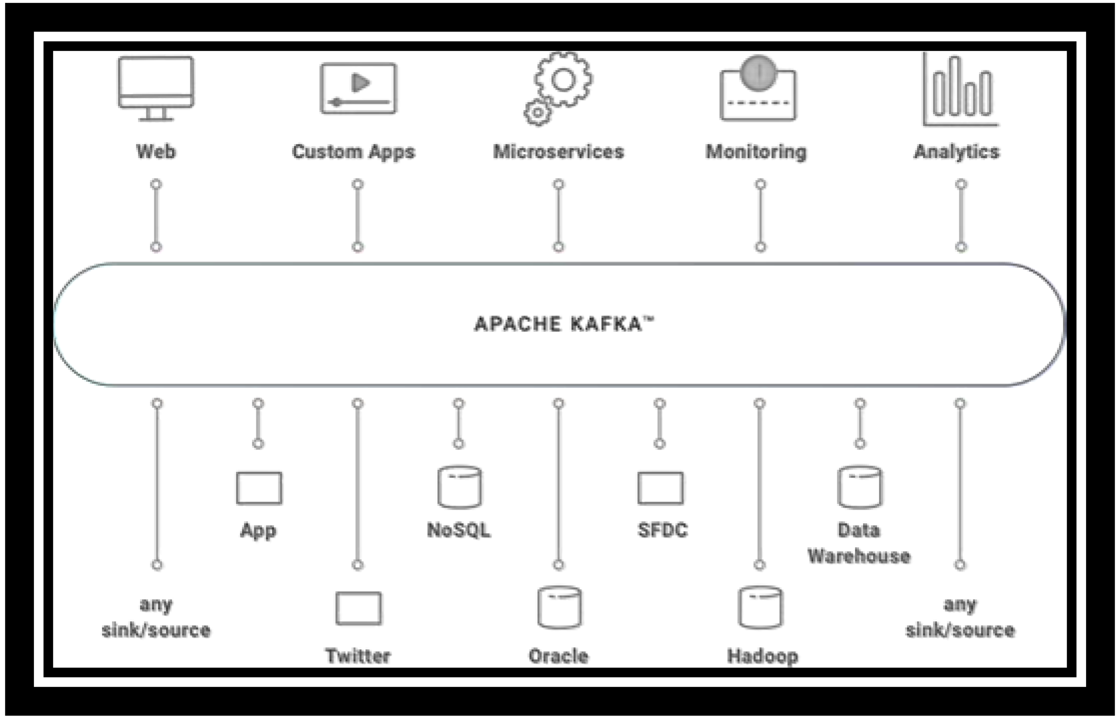


# SO WHAT'S WRONG WITH THIS?

- Technically nothing if it gets the job done, and consumers of the data don't mind getting updated data daily or semi-daily
- But, what if…
  - Processing is taking too long because the batching of data has to happen sequentially for any number of reasons
    - I/O constraints
    - Locking contention due to activity on the primary database
    - Data isn't ready in the remote system until a given time
  - Consumers want near real-time information
  - The primary database from the previous slides are YOUR primary database, so processes now need to be duplicated for other departments

# APACHE KAFKA - A DATA BACKBONE

- "Apache Kafka is a community distributed event streaming platform capable of handling trillions of events a day." https://www.confluent.io/what-is-apache-kafka/
- Trillions of events on Azure https://azure.microsoft.com/en-us/blog/processing-trillions-of-events-per-day-with-apache-kafka-on-azure/
- Originally created as a messaging queue
- A distributed commit log
- Open-sourced by LinkedIn in 2011
- Evolved from  messaging system to a full fledged streaming platform



# HOW'S ABOUT A DEMO?

## WHAT COULD GO WRONG…

# THANK YOU

Allen Underwood

https://www.codingblocks.net

@codingblocks

@theallenu

https://github.com/codingblocks/Batches-to-Streams-with-Apache-Kafka