AWS re:Invent

DEV401

Automating Workflows for Analytics Pipelines

Rob Parrish, Director, Product Management, Treasure Data

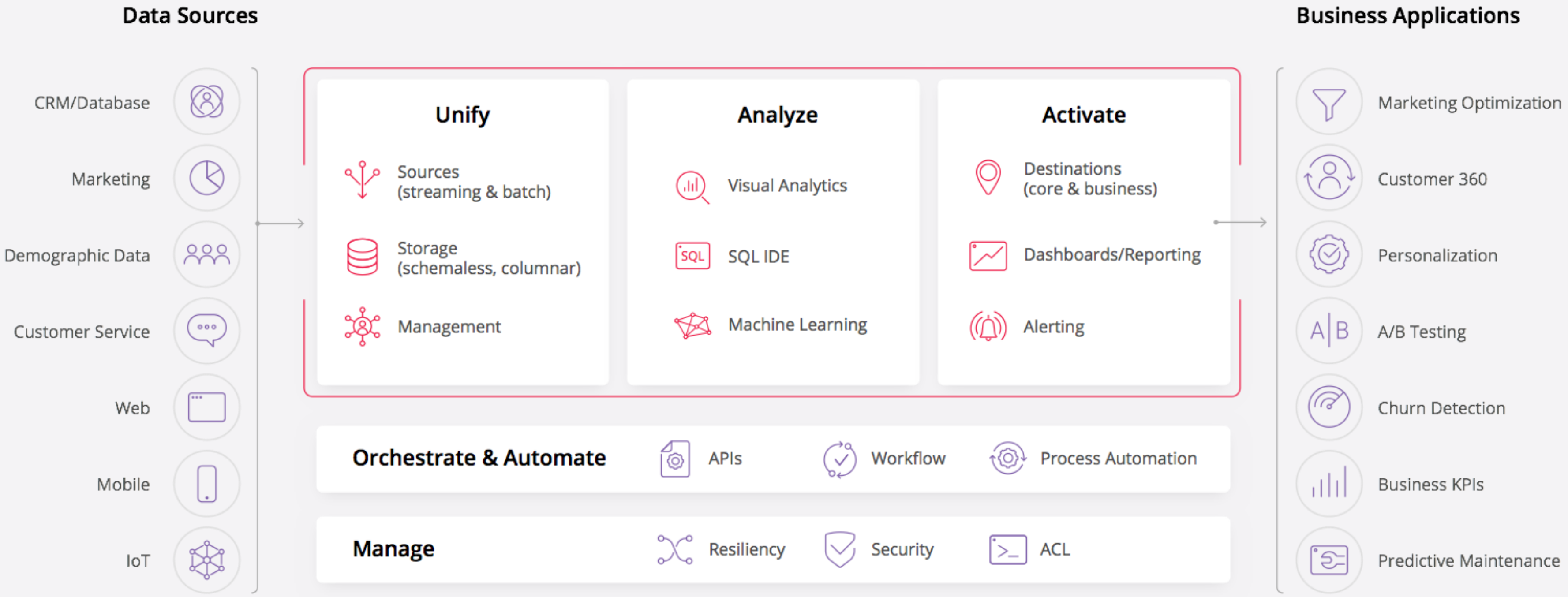
Sadayuki Furuhashi, Founder & Software Architect, Treasure Data

November 29, 2016

amazon web services

TREASURE DATA

Live Data Management Platform



Data Sources

- CRM/Database
- Marketing
- Demographic Data
- Customer Service
- Web
- Mobile
- IoT

Unify

- Sources (streaming & batch)
- Storage (schemaless, columnar)
- Management

Analyze

- Visual Analytics
- SQL IDE
- Machine Learning

Activate

- Destinations (core & business)
- Dashboards/Reporting
- Alerting

Orchestrate & Automate

- APIs
- Workflow
- Process Automation

Manage

- Resiliency
- Security
- ACL

Business Applications

- Marketing Optimization
- Customer 360
- Personalization
- A/B Testing
- Churn Detection
- Business KPIs
- Predictive Maintenance

Treasure Data Background

Founded in 2011 - Headquartered in Silicon Valley (Mountain View, CA)

Global Team: USA, Japan, Korea, India

Innovator in the Data and Analytics OSS Community
Fluentd | Fluent-bit | Embulk | MessagePack | Hivemall | Presto

Key Technology Users and Global Enterprise Customers

Google amazon.com Microsoft GE Healthcare LEXUS

citi VISA SONY LOCKHEED MARTIN J.P.Morgan COMCAST SoftBank

MATOMY MEDIA GROUP PUBLICIS GROUPE dentsu HAKUHODO The New York Times SUBARU MetLife

TREASURE DATA + amazon web services

The Challenge: Managing Data Across Multiple Components

Processing steps managed via CRON

Data transfer require smart retrying

Some processing should only start once data is available

Other processing flows involve 100s of steps

Collaboration is hard, because logic is kept in scripts

It’s particularly hard when data engineers & analysts try to collaborate

Amazon Aurora

Amazon Redshift

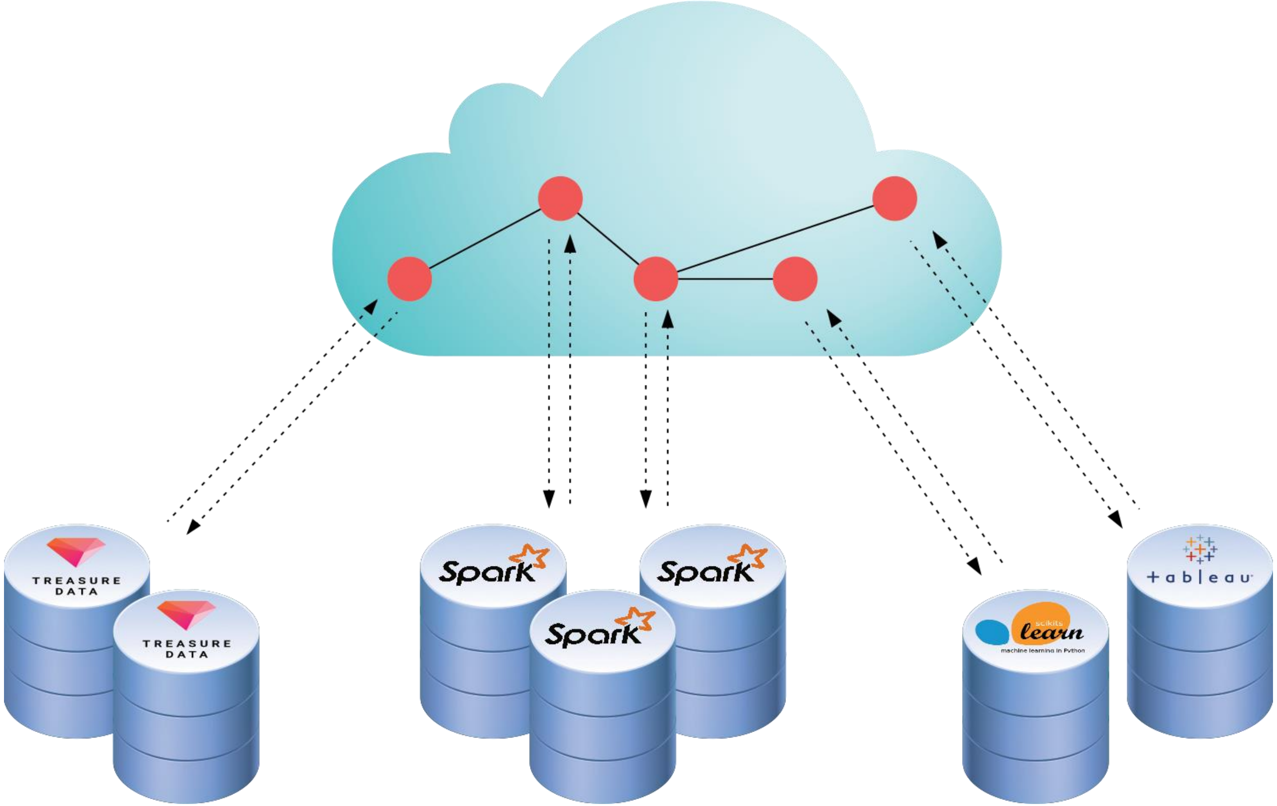
Amazon S3

Amazon EMR

TREASURE DATA

MySQL

Workflows


TREASURE DATA
TREASURE DATA
Spark
Spark
Spark
scikit learn
machine learning in Python
+ableau

Common Steps of Modern Data Processing Workflows

Ingest

Application logs

User attribute data

Ad impressions

3rd-party cookie data

Enrich

Removing bot access

Geo location from IP address

Parsing User-Agent

JOIN user attributes to event logs

Model

A/B Testing

Funnel analysis

Segmentation analysis

Machine learning

Load

Creating indexes

Data partitioning

Data compression

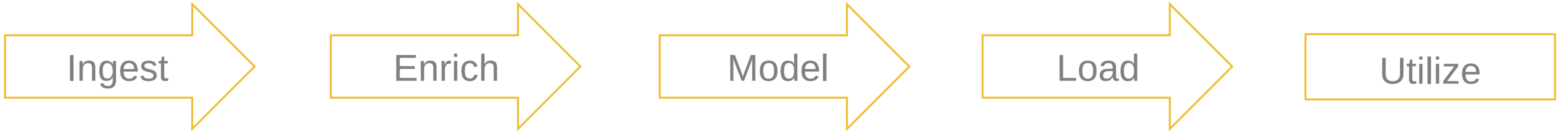
Statistics collection

Utilize

Recommendation API

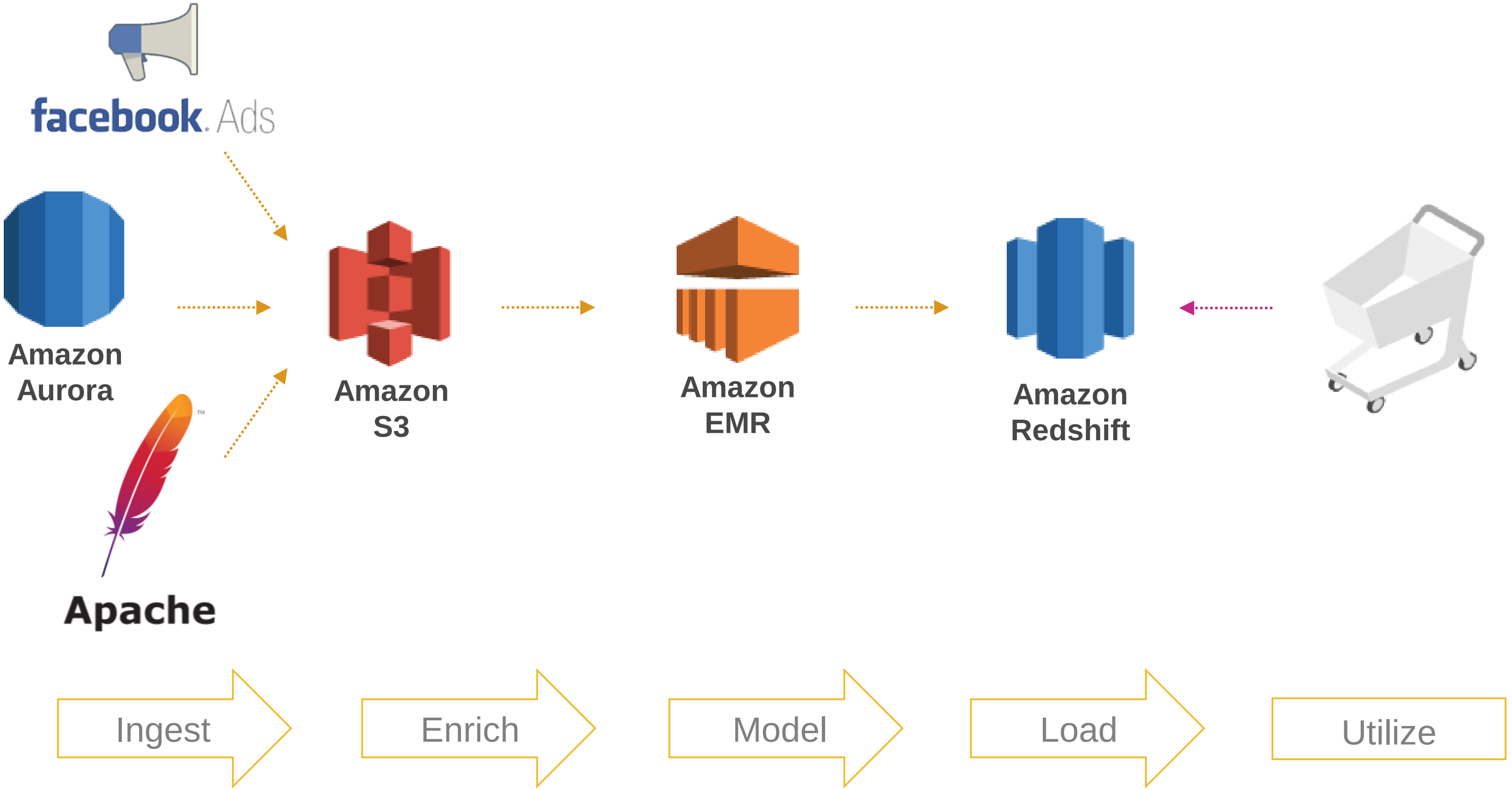
Real-time ad bidding

Visualize using BI applications



Ingest → Enrich → Model → Load → Utilize

Operationalize eCommerce Product Recommendations



facebook. Ads

Amazon Aurora

Apache

Amazon S3

Amazon EMR

Amazon Redshift

Ingest

Enrich

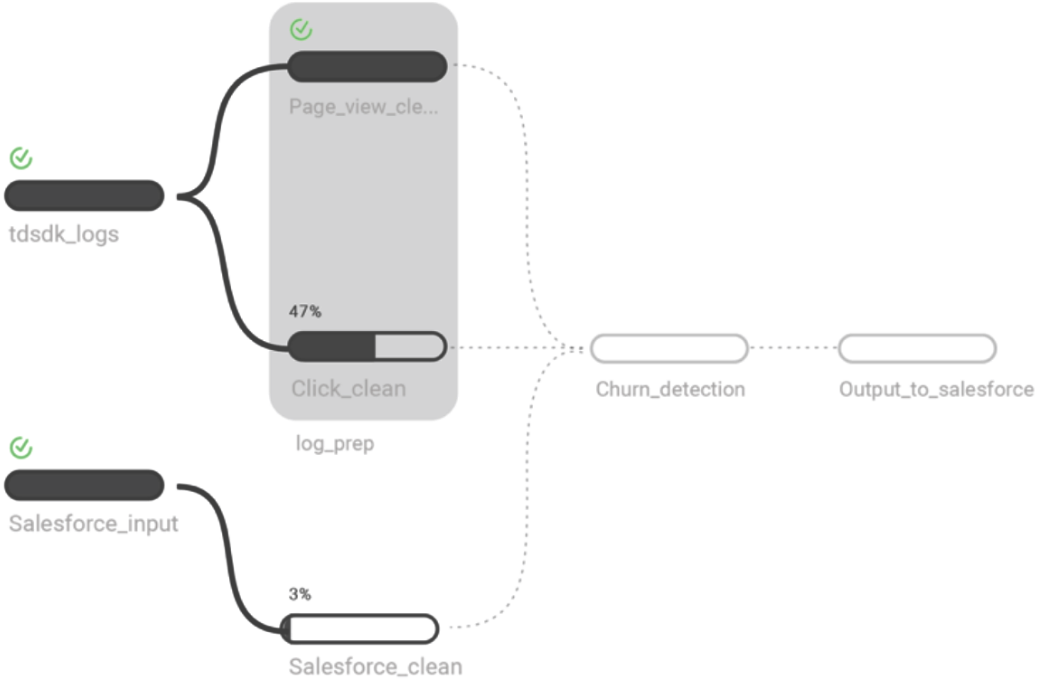
Model

Load

Utilize

Solution: Build a Workflow Tool Everyone can Leverage

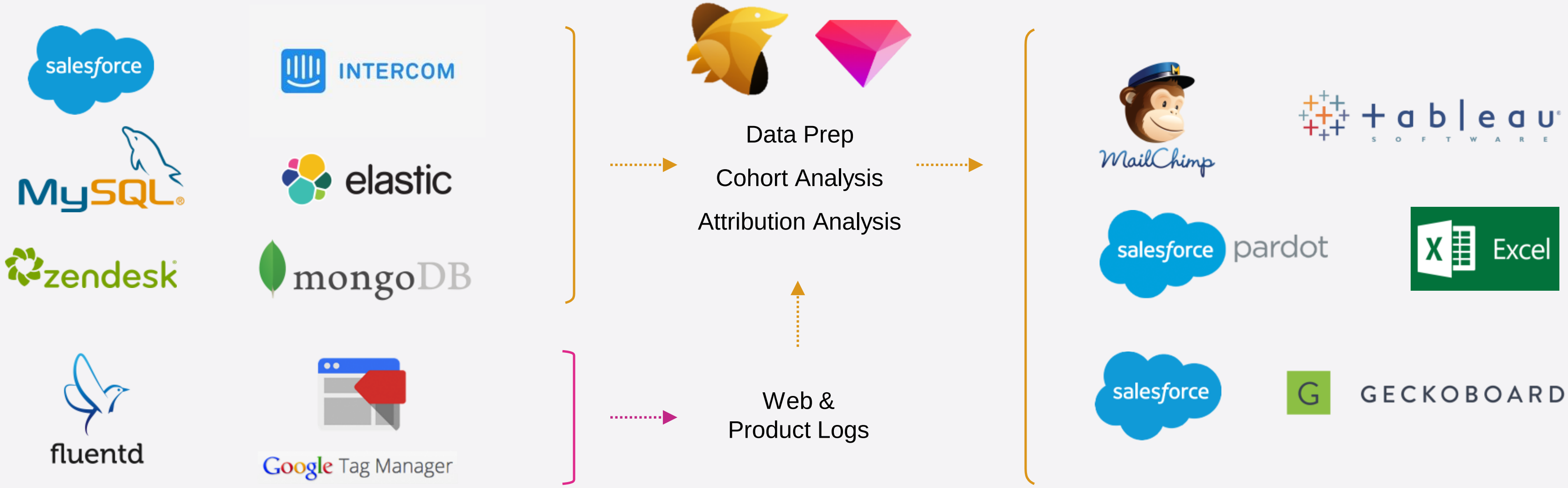
digdag



Packlink®

Packlink is an online platform providing cost-effective package delivery services in Europe & Internationally.

They use Digdag to manage their analytic workflows that power insights that allow Sales, Marketing, and their Partners to operate more effectively – helping their business to grow.



Data Prep

Cohort Analysis

Attribution Analysis

Web & Product Logs

“Using Digdag, I now **feel confident in my ability to manage complex analytic flows**. From ETL processes for transferring data, to analytic steps for running attribution or cohort analysis, to deploying those insights back into the cloud systems my company uses to run our business.

It’s enabled us to get out refreshes of these **insights more timely for our analytic consumers** - sales, marketing, and the executive suite. I now can feel confident each night that our analysis will be completed as expected.”

Pierre Bèvillard
Head of Business Intelligence

Sadayuki Furuhashi

A founder of Treasure Data.
An open-source hacker.
github: @frsyuki

MessagePack

It’s like JSON, but fast and small

fluentd

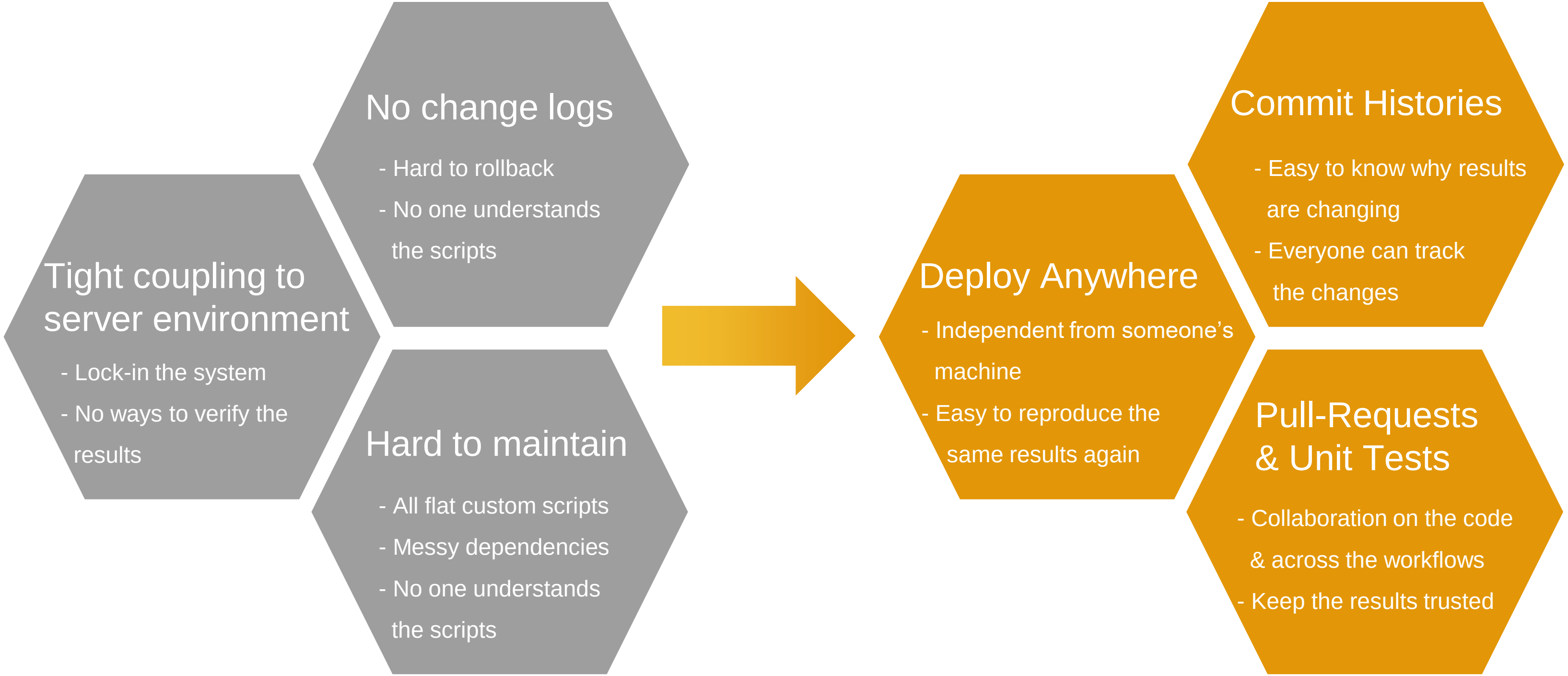
Unified log collection infrastructure

embulk

Plugin-based ETL tool

Why Digdag?

Bringing Best Practices of Software Development



Tight coupling to server environment

- Lock-in the system
- No ways to verify the results

No change logs

- Hard to rollback
- No one understands the scripts

Hard to maintain

- All flat custom scripts
- Messy dependencies
- No one understands the scripts

Deploy Anywhere

- Independent from someone's machine
- Easy to reproduce the same results again

Commit Histories

- Easy to know why results are changing
- Everyone can track the changes

Pull-Requests & Unit Tests

- Collaboration on the code & across the workflows
- Keep the results trusted

Encourage Use of Application Development Best Practices

Parameterized Modules

- redshift>
- emr>
- …

Rather than one giant unwieldy script, break queries into manageable, well-identified modules to aid in collaboration, updates and maintenance.

Task Grouping

- From bird’s eye to details

Enable query writers to specify dependencies easily, without having to slog through hundreds of lines of code to make a change.

Automated Validation

- Verify results between steps

Automate validation of intermediate data to encourage testing of data results over time. As data changes, we can ensure we know. We can keep the results always trusted.

Unite Engineering & Analytic Teams

Powerful for Engineers

Our goal is to make it feasible for our most advanced users to take advantage of engineering teams to manage using their favorite tools (e.g. git).

Friendly for Analysts

While, also making the definition file straight forward enough for a wider range of analysts to leverage & use

```
_export:
  td:
    database: workflow_temp

+task1:
  td>: queries/daily_open.sql
  create_table: daily_open
+task2:
  td>: queries/monthly_open.sql
  create_table: monthly_open
```

Workflow Constructs

Operators

Standard libraries

redshift>: runs Amazon Redshift queries
emr>: create/shutdowns a cluster & runs steps
s3_wait>: waits until a file is put on S3
pg>: runs PostgreSQL queries
td>: runs Treasure Data queries
td_for_each>: repeats task for result rows
mail>: sends an email

Open-source libraries

You can release & use open-source operator libraries.

```
+wait_for_arrival:
  s3_wait>: |
   bucket/www_${date}.csv

+load_table:
  redshift>: scripts/copy.sql
```

Scripting operators

Scripting operators

sh>: runs a Shell script
py>: runs a Python method
rb>: runs a Ruby method

Docker option

```
docker:
  image: ubuntu:16.04
```

Digdag supports Docker natively.
Easy to use data analytics tools.
Reproducible anywhere.

```
+run_custom_script:
  sh>: scripts/custom_work.sh

+run_python_in_docker:
  py>: Analysis.item_recommends
  docker:
    image: ubuntu:16.04
```

Loops and parameters

Parameter

A task can propagate parameters to following tasks

Loop

Generate subtasks dynamically so that Digdag applies the same set of operators to different data sets.

```
+send_email_to_active_users:
  td_for_each>: list_active.sql
  _do:
   +send:
     email>: tempalte.txt
     to: ${td.for_each.addr}
```

(+send tasks are dynamically generated)

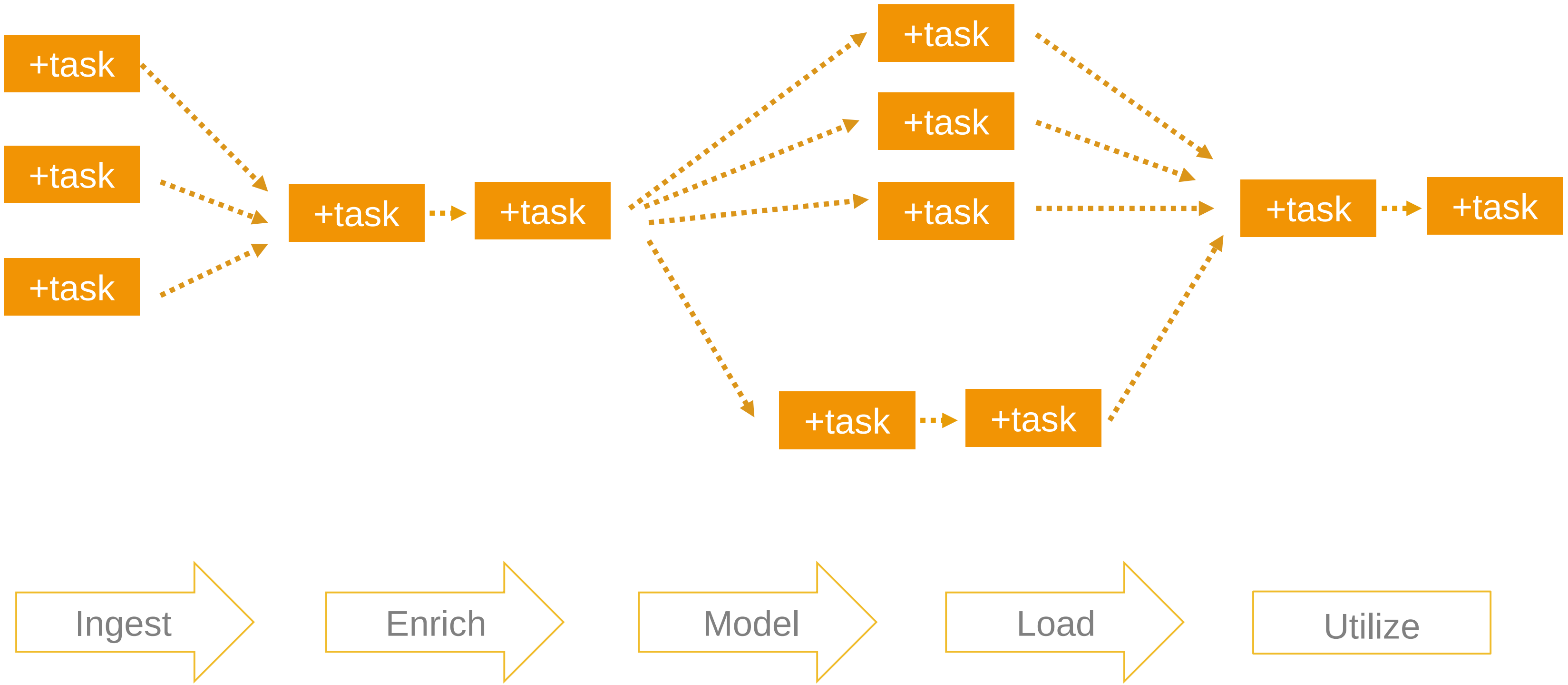
Parallel execution

Parallel execution

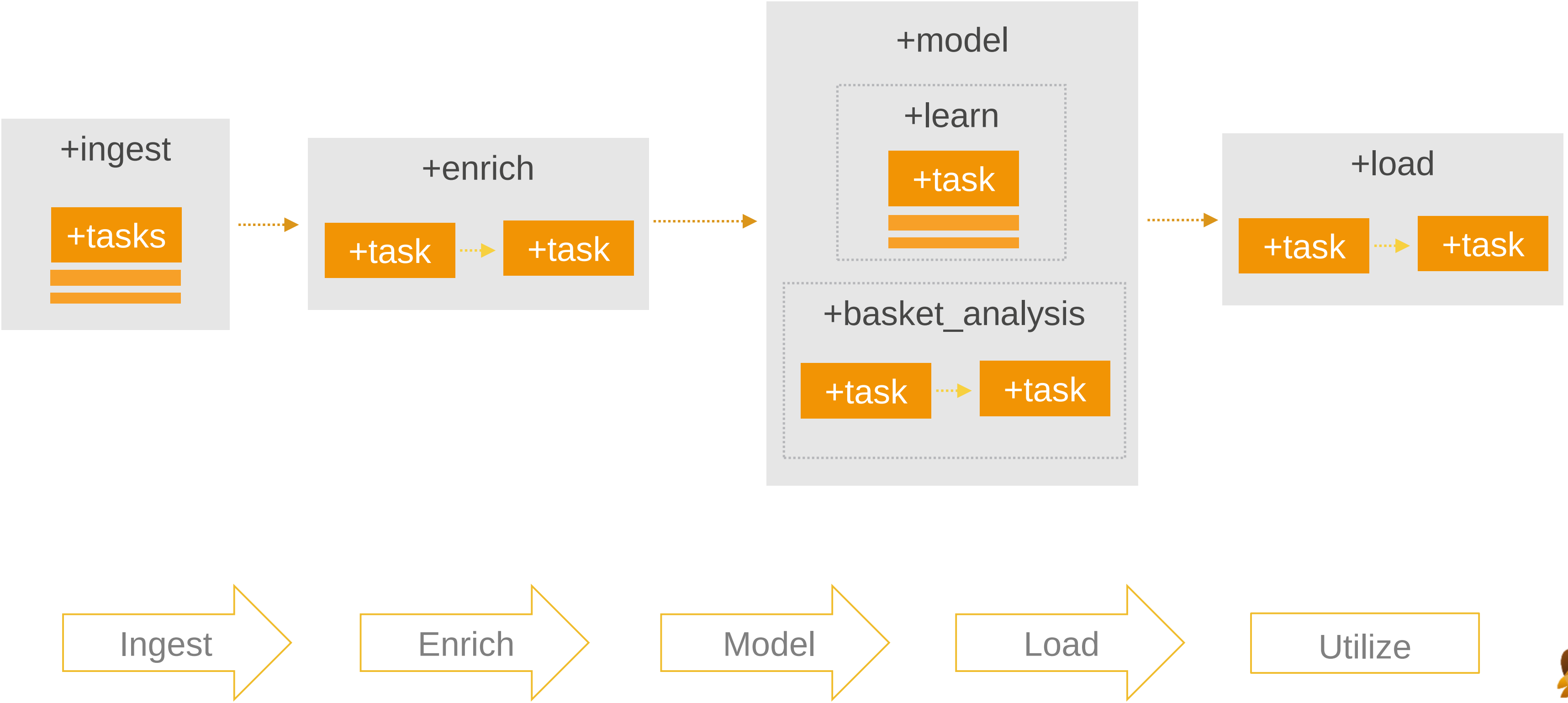
Tasks under a same group run in parallel if _parallel option is set to true.

```
+load_data:
  _parallel: true

  +load_users:
    redshift>: copy/users.sql

  +load_items:
    redshift>: copy/items.sql
```

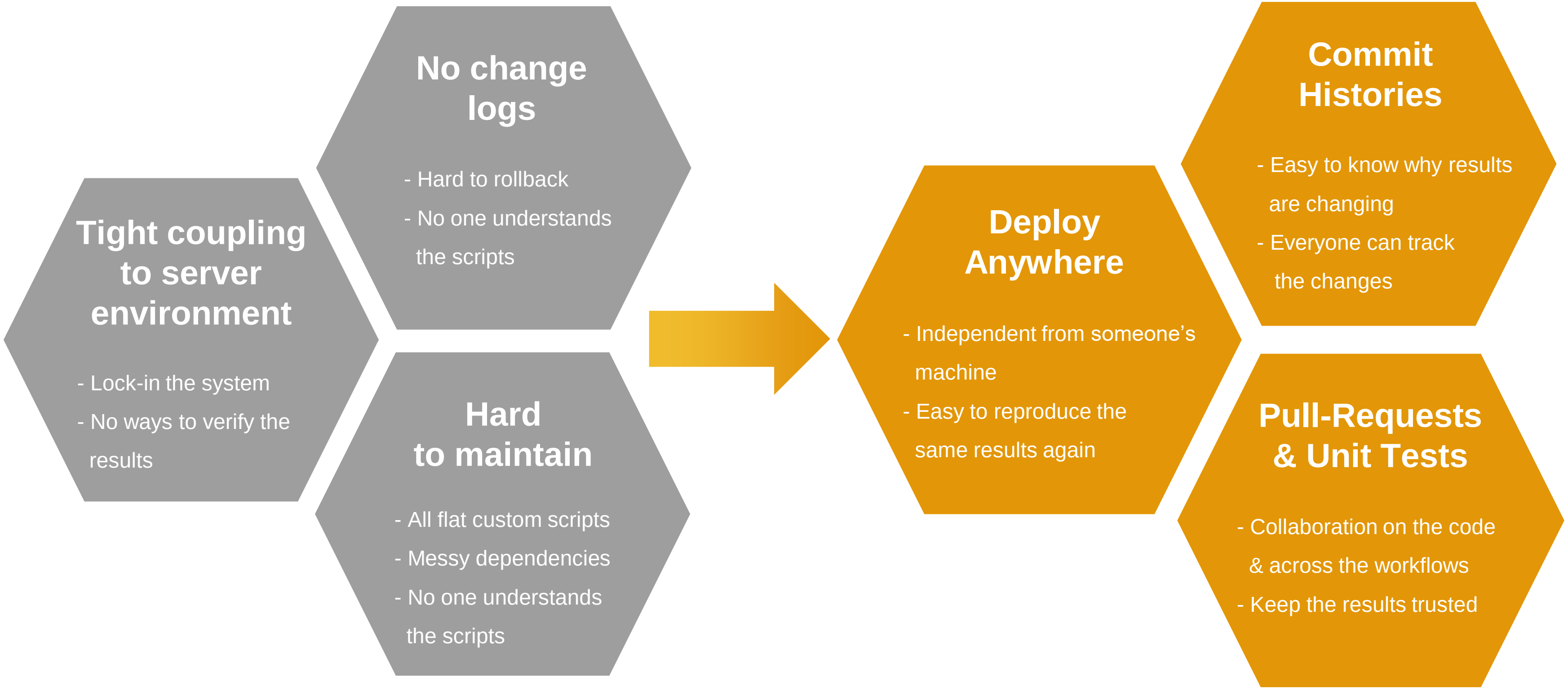
Workflow Steps



Organizing tasks using groups



Bringing Best Practices of Software Development



Tight coupling to server environment

- Lock-in the system
- No ways to verify the results

No change logs

- Hard to rollback
- No one understands the scripts

Hard to maintain

- All flat custom scripts
- Messy dependencies
- No one understands the scripts

Deploy Anywhere

- Independent from someone’s machine
- Easy to reproduce the same results again

Commit Histories

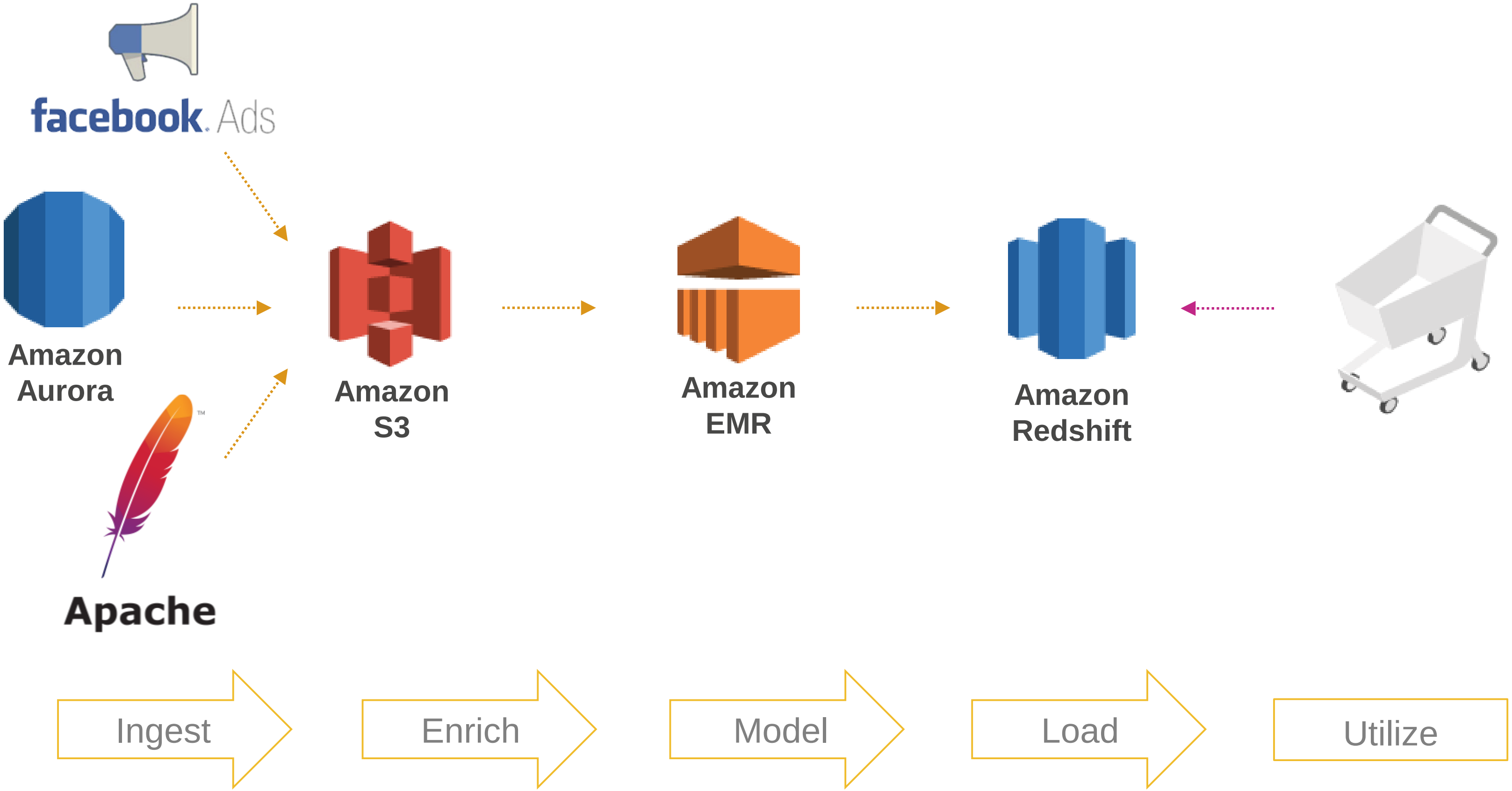
- Easy to know why results are changing
- Everyone can track the changes

Pull-Requests & Unit Tests

- Collaboration on the code & across the workflows
- Keep the results trusted

Demo

Operationalize eCommerce Product Recommendations



facebook. Ads

Amazon Aurora

Apache

Amazon S3

Amazon EMR

Amazon Redshift

Ingest

Enrich

Model

Load

Utilize

to the demo

Conclusion

Digdag Supports our Customers

digdag

TREASURE DATA

Scheduling

AWS System Processing

Query Result Output

Loading Bulk Data

Presto Analytic Queries

ETL Process Management

Built to Handle Production Workloads

Managing Cloud Infrastructure

It’s not easy! The lessons we learn are always applied to our OSS for the good of the community.

Maintaining 24/7 Uptime

With the complexities of the modern data stack, what we need is Continuous Data Integration.

Handling over 100 Billion Queries

Ensuring robust operation with scale is a huge issue for us.

AWS re:Invent

Thank you!

digdag

www.digdag.io

www.treasuredata.com

Visit our booth #1818 for more info,
and for VIP wristbands to our party at TAO tonight!

Remember to complete your evaluations!