Modern Storages and Data Warehousing
Week 7 - Advanced ETL

Попов Илья, i.popov@hse.ru

Recap прошлых занятий

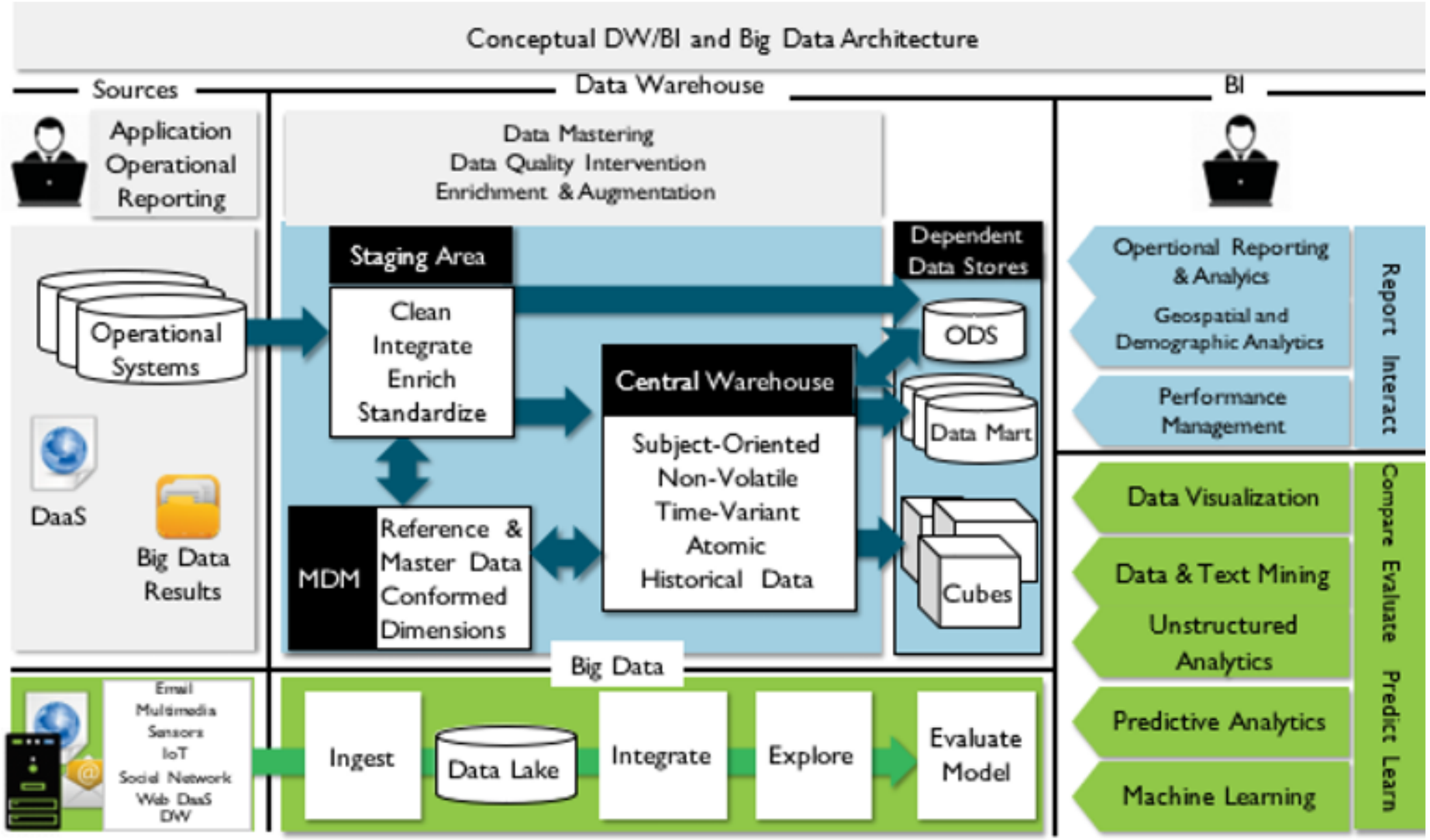

Figure 5: Date Warehouse Concept

1 - Шаблонизация

Мотивация

〉 Иногда мы хотим, чтобы наш код генерировался динамически
〉 Для этого мы хотим написать шаблон, и потом подставлять в него куски кода в зависимости от каких-то условий / переменных среды

Jinja

- Выпущен в 2008 году
- Сразу писалась командой Pallets Project как opensource проект
- Написан на Python
- Похож на стандартный шаблонизатор django

Обращение к переменным и функциям

```
>>> Template("{{ var }}").render(var=12)
'12'
>>> Template("{{ var }}").render(var="hello")
'hello'
>>>
```

```
>>> def foo():
...     return "foo() called"
...
>>>
>>> Template("{{ foo() }}").render(foo=foo)
'foo() called'
>>>
```

Циклы и условия

```
{% if user %}
    {% if user.newbie %}
        <p>Display newbie stages</p>
    {% elif user.pro %}
        <p>Display pro stages</p>
    {% elif user.ninja %}
        <p>Display ninja stages</p>
    {% else %}
        <p>You have completed all states</p>
    {% endif %}
{% else %}
    <p>User is not defined</p>
{% endif %}
```

```
{% set user_list = ['tom', 'jerry', 'spike'] %}

<ul>
{% for user in user_list %}
    <li>{{ user }}</li>
{% endfor %}
</ul>
```

А еще…

- Можно писать свои функции - макросы
- Можно наследовать и вкладывать шаблоны друг в друга
- Можно применять фильтры и модифицировать множества до рендера

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}Default Title{% endblock %}</title>
</head>
<body>

    {% block nav %}
        <ul>
            <li><a href="/home">Home</a></li>
            <li><a href="/api">API</a></li>
        </ul>
    {% endblock %}

    {% block content %}

    {% endblock %}
</body>
</html>
```

```
{% extends 'base.html' %}



{% block content %}
    {% for bookmark in bookmarks %}
        <p>{{ bookmark.title }}</p>
    {% endfor %}
{% endblock %}
```

А зачем?

- В основном - нужно web-dev’ам
- Но Jinja де-факто стала стандартом в кодогенерации
- Это знание нам нужно для того, чтобы понимать генерацию в Airflow и dbt

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}Default Title{% endblock %}</title>
</head>
<body>

    {% block nav %}
        <ul>
            <li><a href="/home">Home</a></li>
            <li><a href="/api">API</a></li>
        </ul>
    {% endblock %}

    {% block content %}

    {% endblock %}
</body>
</html>
```

```
{% extends 'base.html' %}


{% block content %}
    {% for bookmark in bookmarks %}
        <p>{{ bookmark.title }}</p>
    {% endfor %}
{% endblock %}
```

2 - Продвинутые сценарии Airflow (live demo)

3 - dbt

Мотивация

〉 Хотим навести порядок в наших ETL-процессах
〉 Хотим упростить процедуру миграции данных
〉 Хотим дополнительные плюшки - data lineage и документацию
〉 Хотим инструмент, который позволит создавать и тестировать миграции в разных окружениях

dbt

- Data Build Tools
- Относительно новый проект (буквально несколько лет)
- Недавно привлек $200M инвестиций
- Jinja-шаблонизатор над ETL
- Позволяет делать кучу интересностей

dbt_

dbt & automate DV
(live demo)