CSCI-E-29

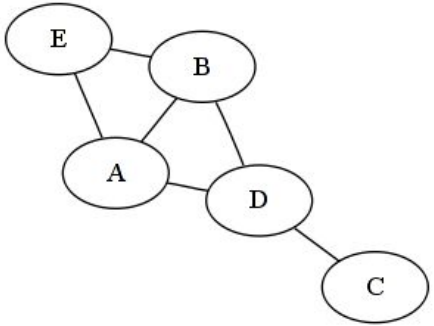
Fall 2019

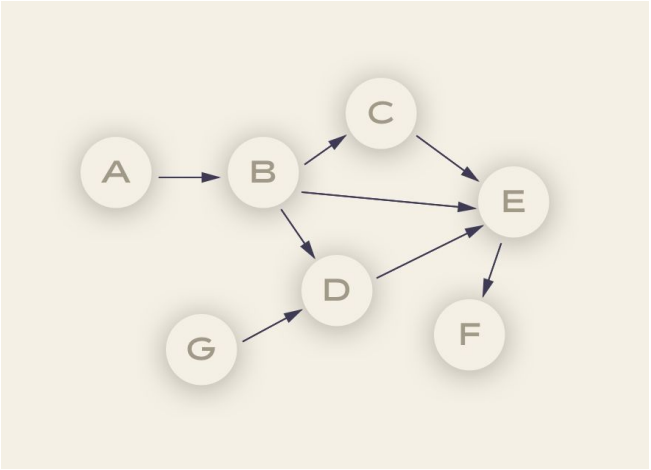
Topics

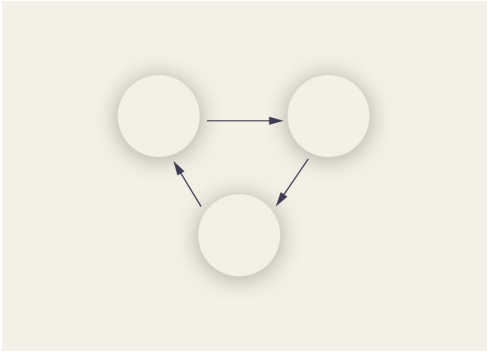
- Luigi
- Demo
- Q&A

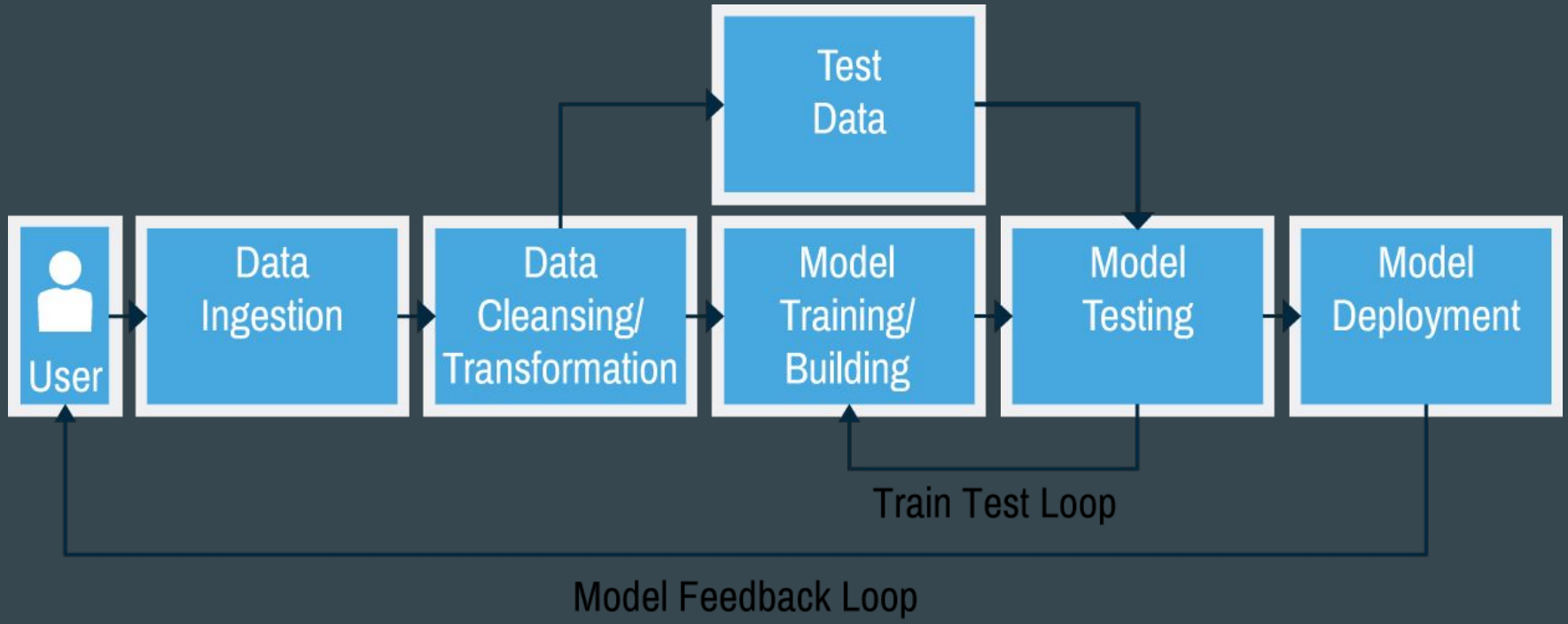
Directed Acyclic Graphs (DAGs)

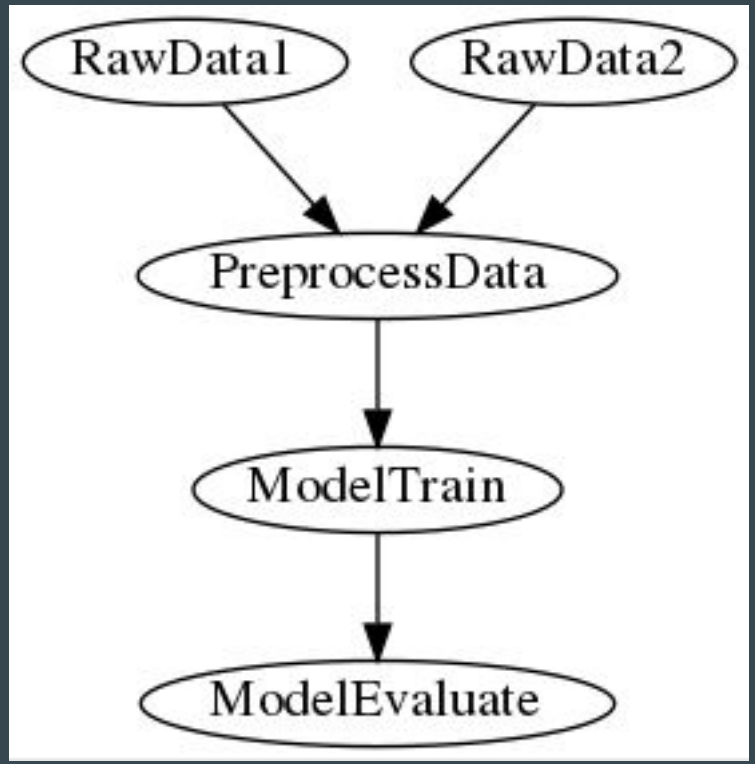
- Directed vs undirected graphs
- Acyclic vs cyclic graphs

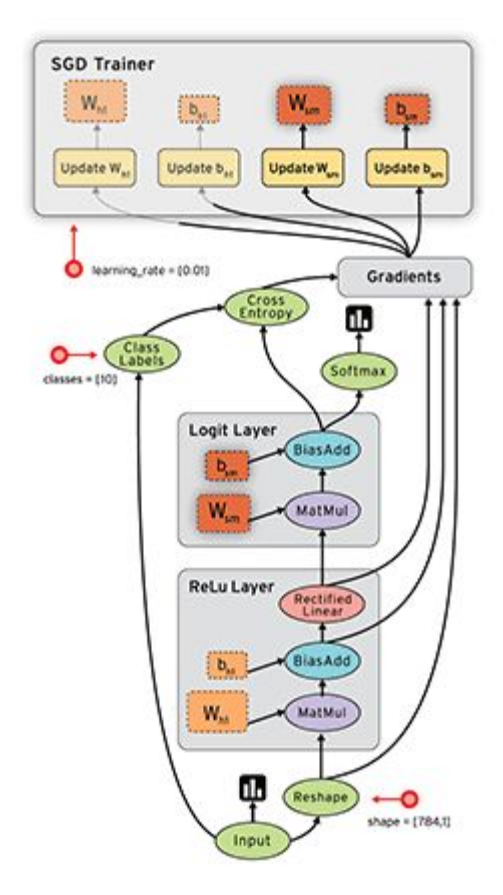


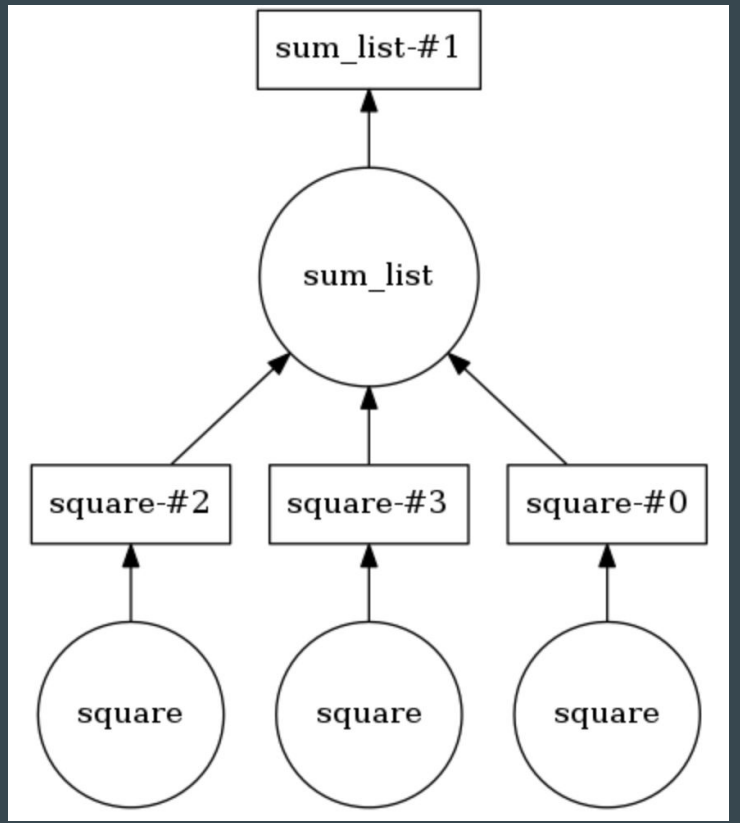




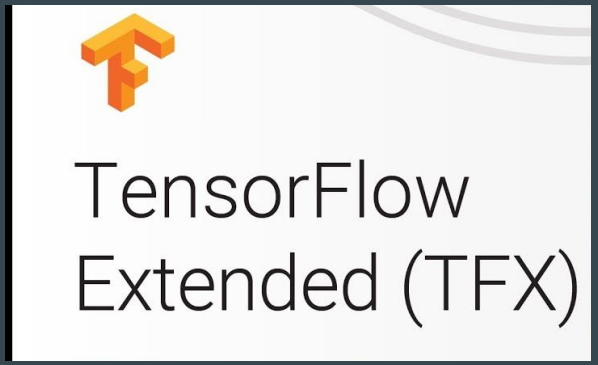

Test
Data
User
Data
Ingestion
Data
Cleansing/
Transformation
Model
Training/
Building
Model
Testing
Model
Deployment
Train Test Loop
Model Feedback Loop


RawData1
RawData2
PreprocessData
ModelTrain
ModelEvaluate


SGD Trainer
Update W_hl
Update b_hl
Update W_sm
Update b_sm
learning_rate = [0.01]
Gradients
Cross Entropy
Class Labels
classes = [10]
Softmax
Logit Layer
b_sm
BiasAdd
W_sm
MatMul
ReLu Layer
Rectified Linear
b_hl
BiasAdd
W_hl
MatMul
Reshape
shape = [784,1]
Input

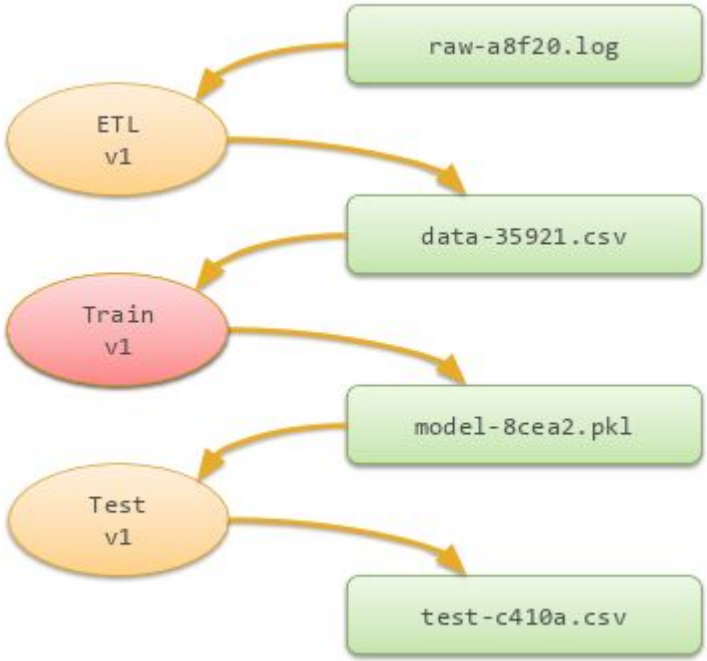

sum_list-#1
sum_list
square-#2
square-#3
square-#0
square
square
square

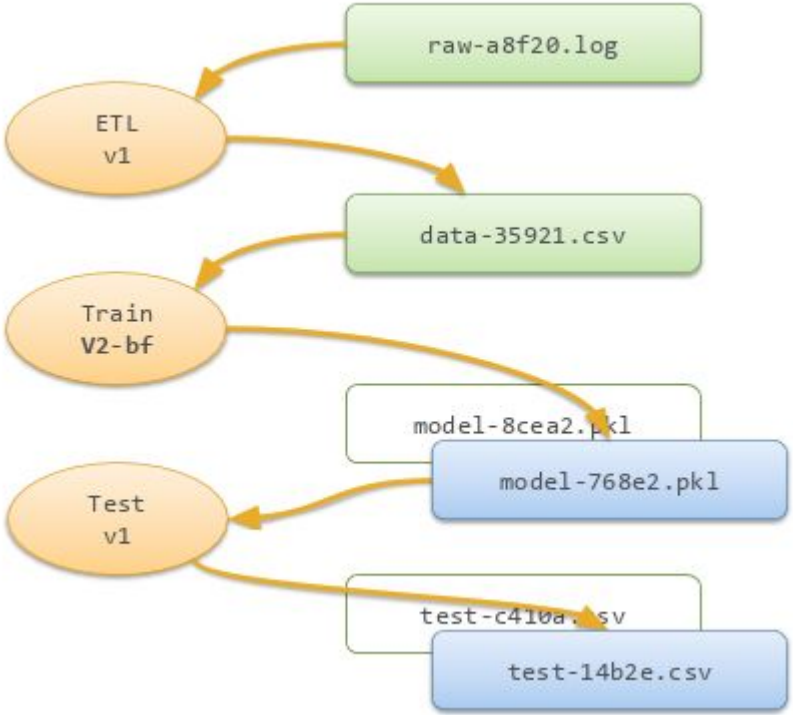
Kubeflow



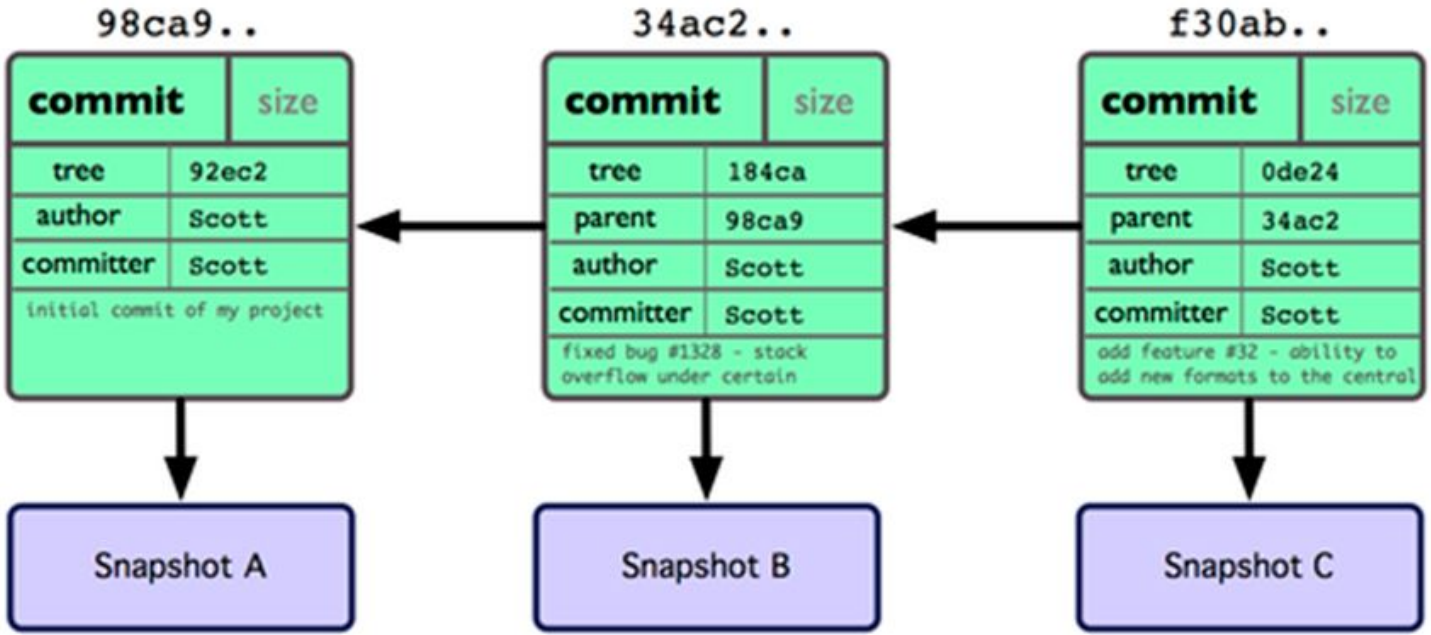
Salted

Salted: Graphs

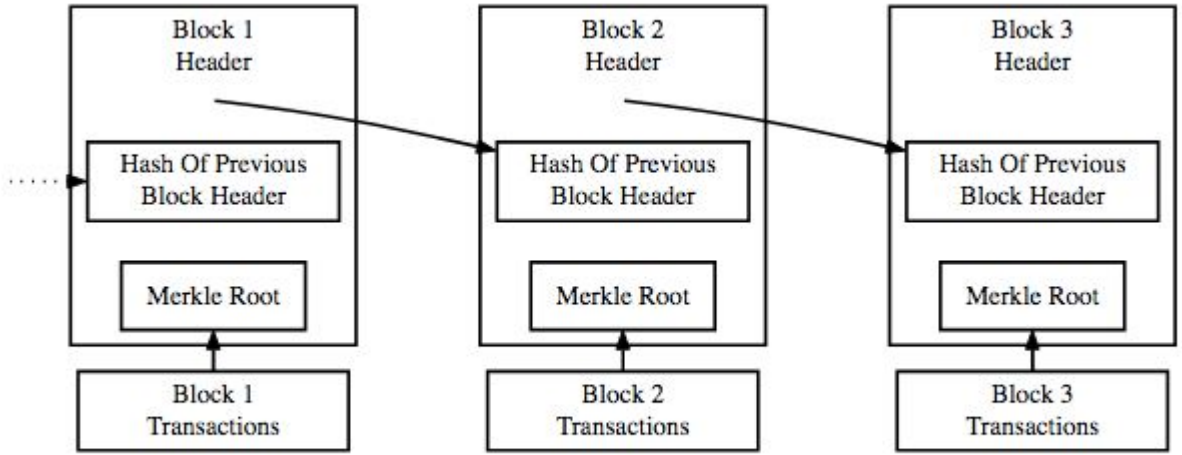




Salted: Git


98ca9..
commit
size
tree
92ec2
author
Scott
committer
Scott
initial commit of my project
34ac2..
commit
size
tree
184ca
parent
98ca9
author
Scott
committer
Scott
fixed bug #1328 - stack overflow under certain
f30ab..
commit
size
tree
0de24
parent
34ac2
author
Scott
committer
Scott
add feature #32 - ability to add new formats to the central
Snapshot A
Snapshot B
Snapshot C

Salted: Blockchain



Example

Spotify:

- What were the top 10 most popular artists in 2017 (i.e. most number of song plays)
- Stream data is stored in log text files, one for each day. Each row represents a stream (timestamp, artist, track)

Descriptors and Composition

Descriptors

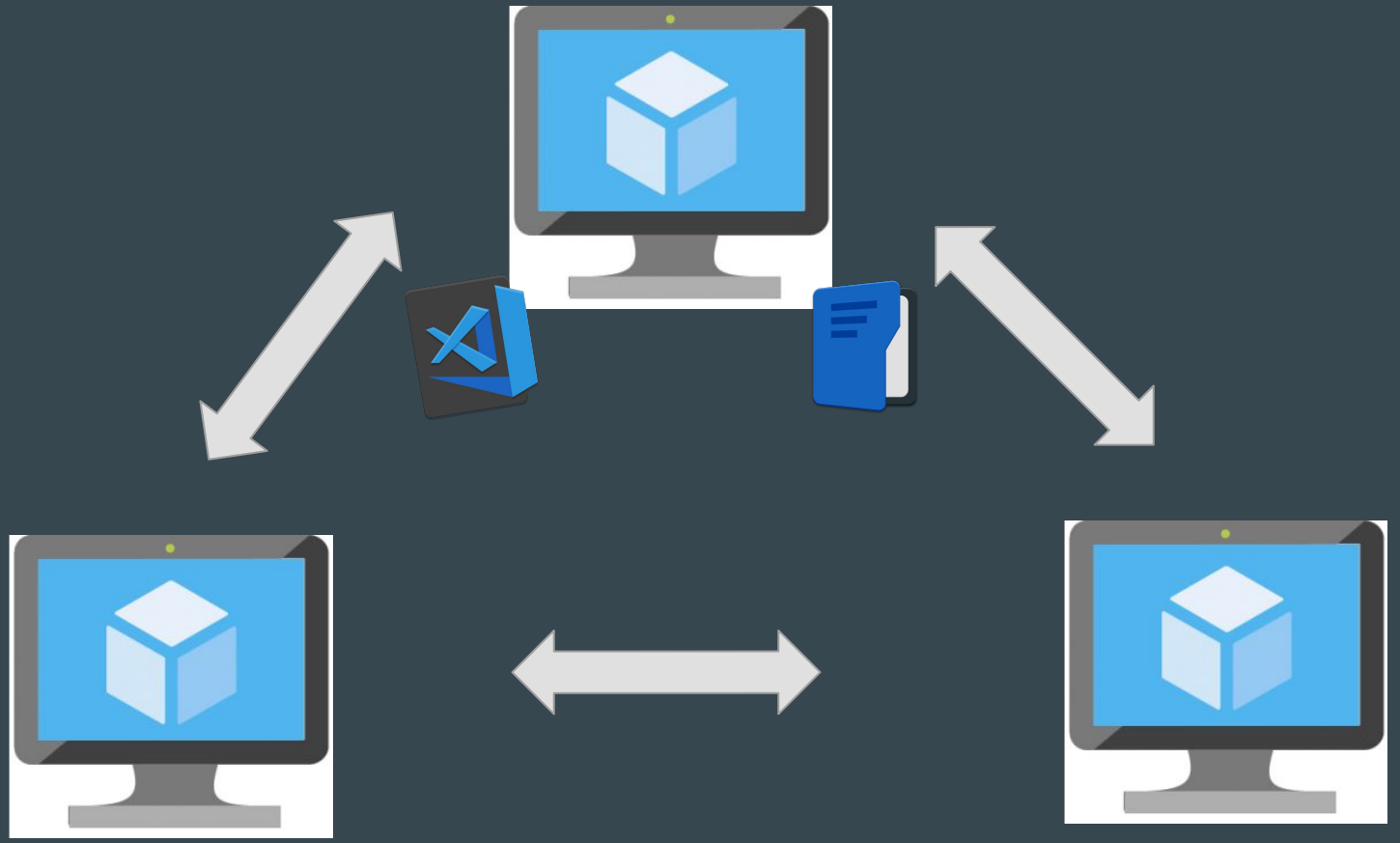
In general, a descriptor is an object attribute with “binding behavior”, one whose attribute access has been overridden by methods in the descriptor protocol. Those methods are `__get__()`, `__set__()`, and `__delete__()`. If any of those methods are defined for an object, it is said to be a descriptor.

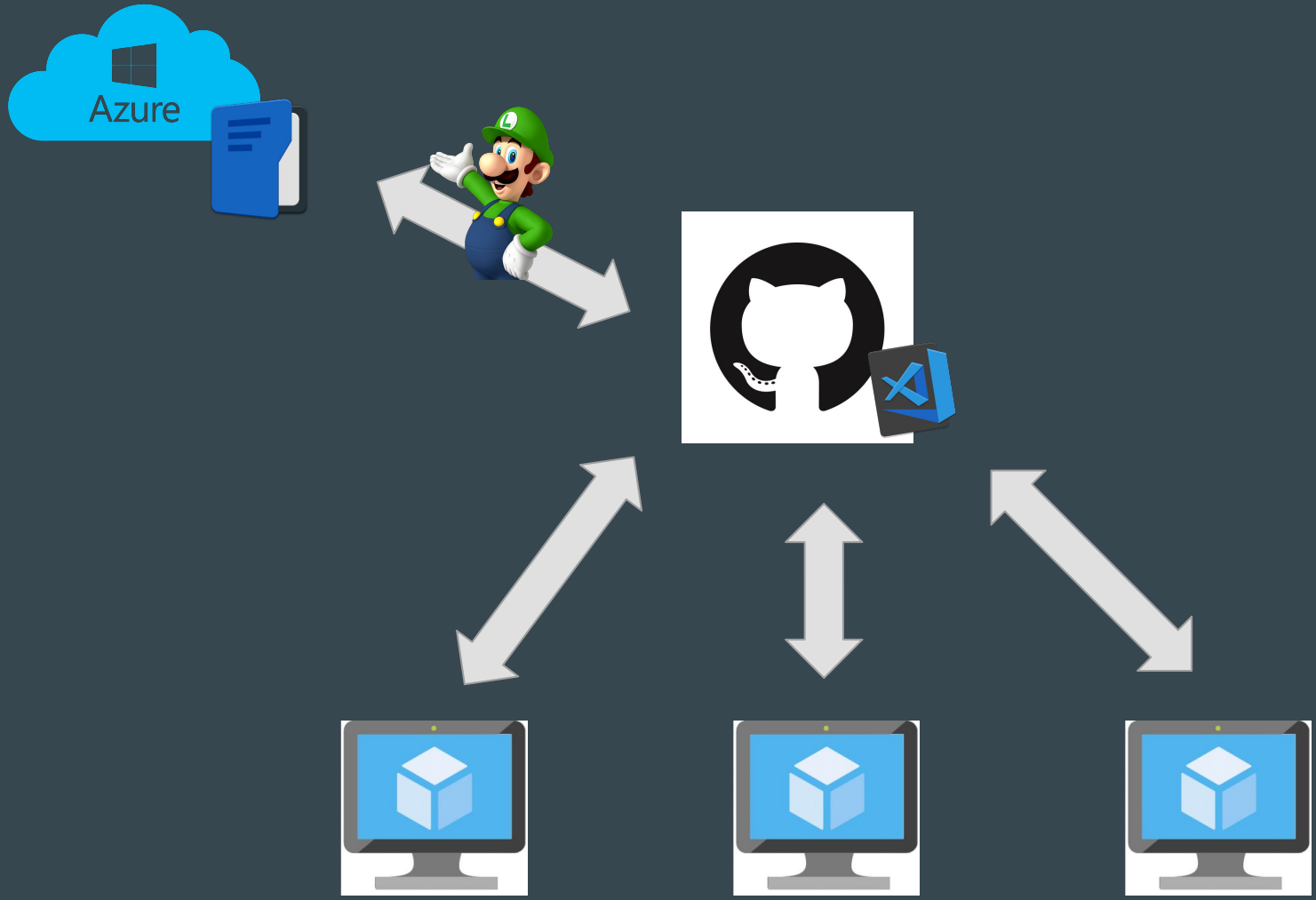
Descriptors

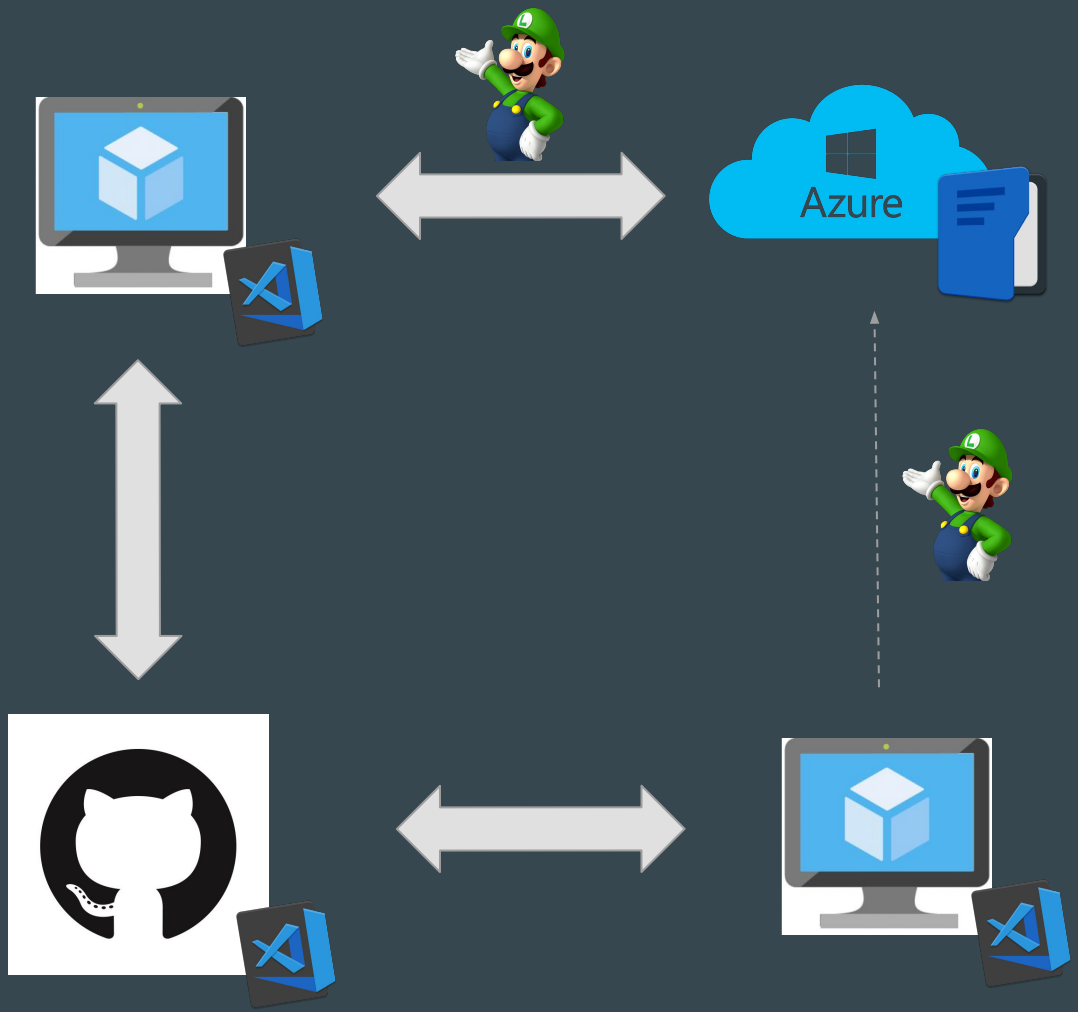
```
class RevealAccess(object):
    """A data descriptor that sets and returns values
       normally and prints a message logging their access.
    """

    def __init__(self, initval=None, name='var'):
        self.val = initval
        self.name = name

    def __get__(self, obj, objtype):
        print('Retrieving', self.name)
        return self.val

    def __set__(self, obj, val):
        print('Updating', self.name)
        self.val = val
```

```
>>> class MyClass(object):
...     x = RevealAccess(10, 'var "x"')
...     y = 5
...
>>> m = MyClass()
>>> m.x
Retrieving var "x"
?
>>> m.x = 20
Updating var "x"
>>> m.x
Retrieving var "x"
?
>>> m.y
?
```

Why build pipelines?




Azure


Azure

Q&A