CSCI-E-29 Session: Week 6

October 2018

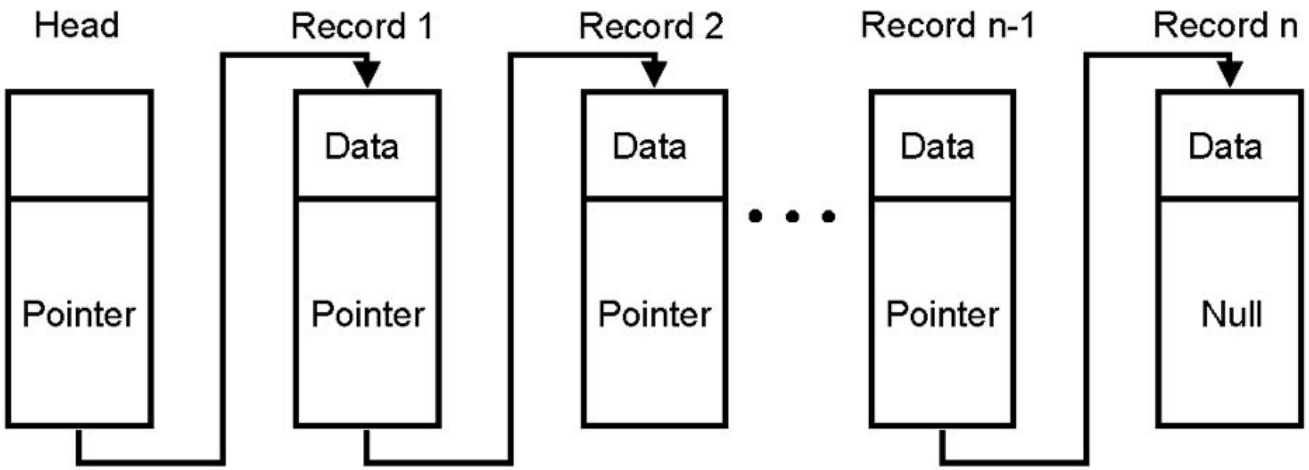
Topics

- Discussion: Pset-2 (5m)
- Salted (20m)
- Descriptors (10m)
- Composition (15m)
- Q&A (10)

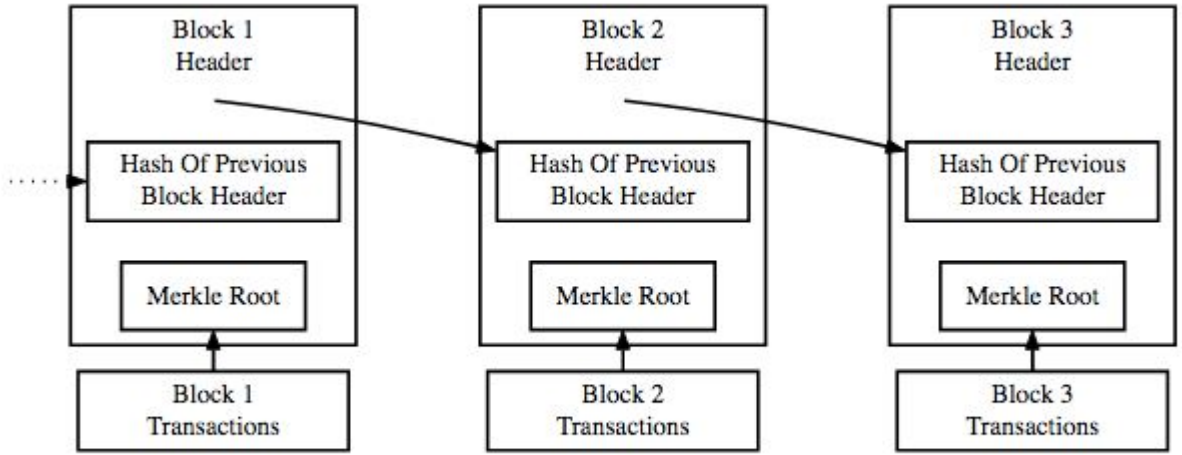
Discussion: Pset-2

Salted

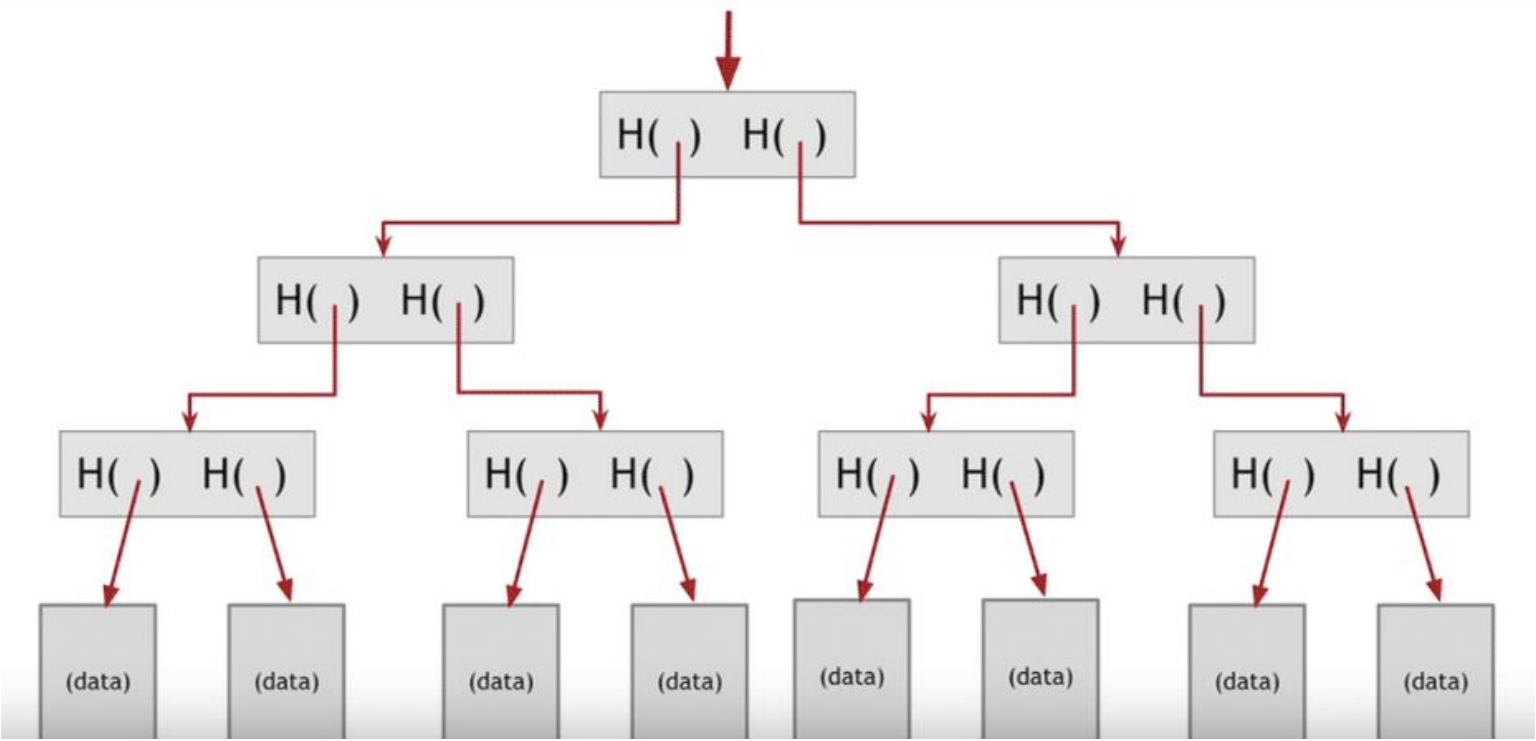
Salted: Blockchain



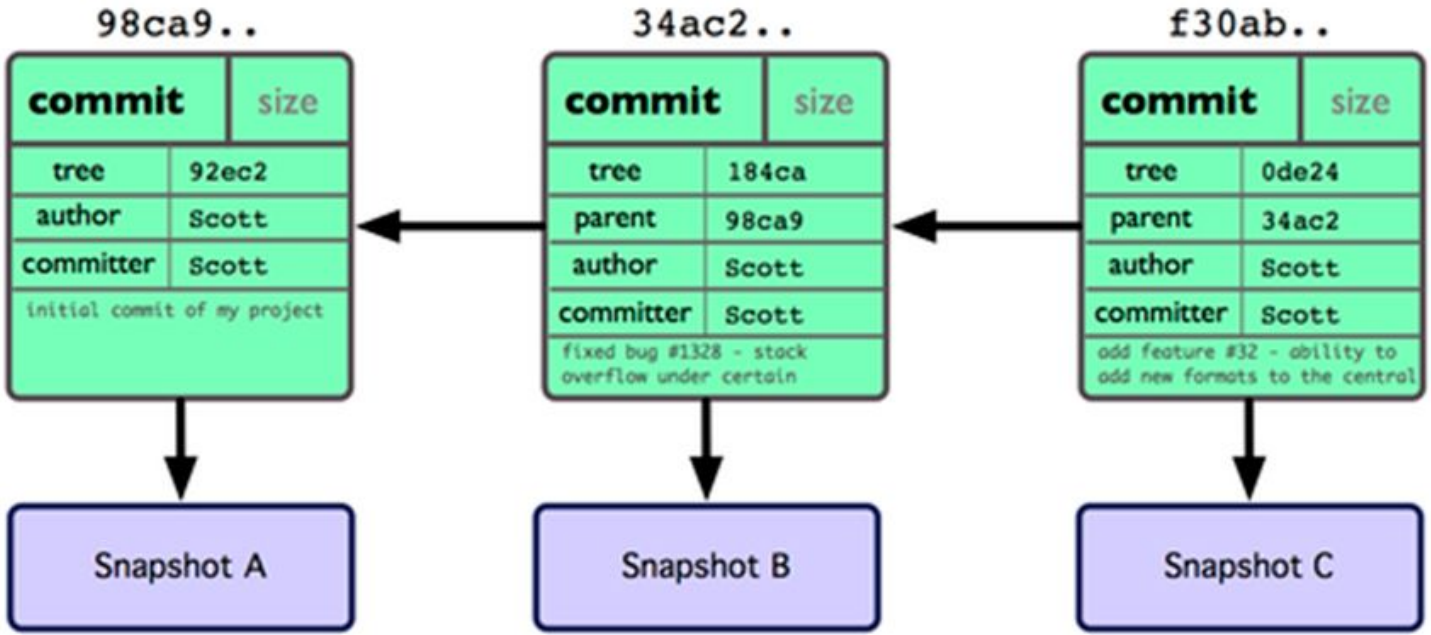
Salted: Blockchain



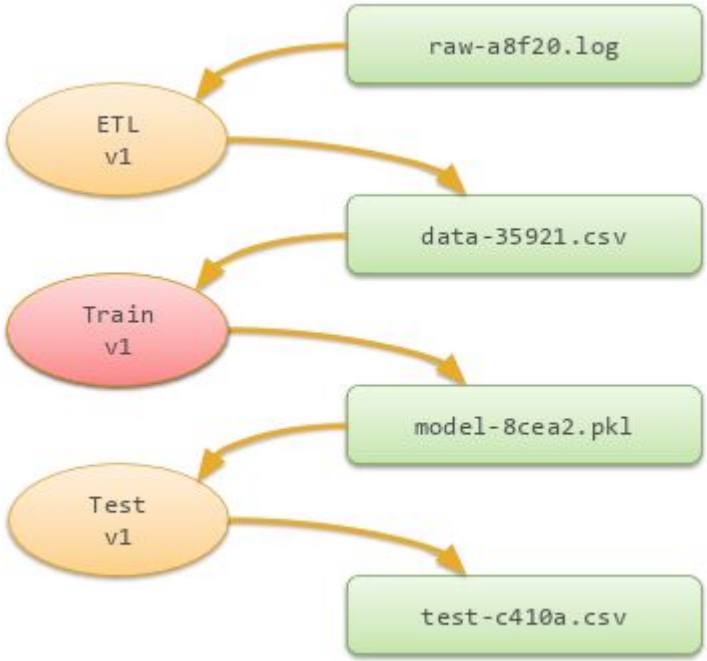
Salted: Blockchain

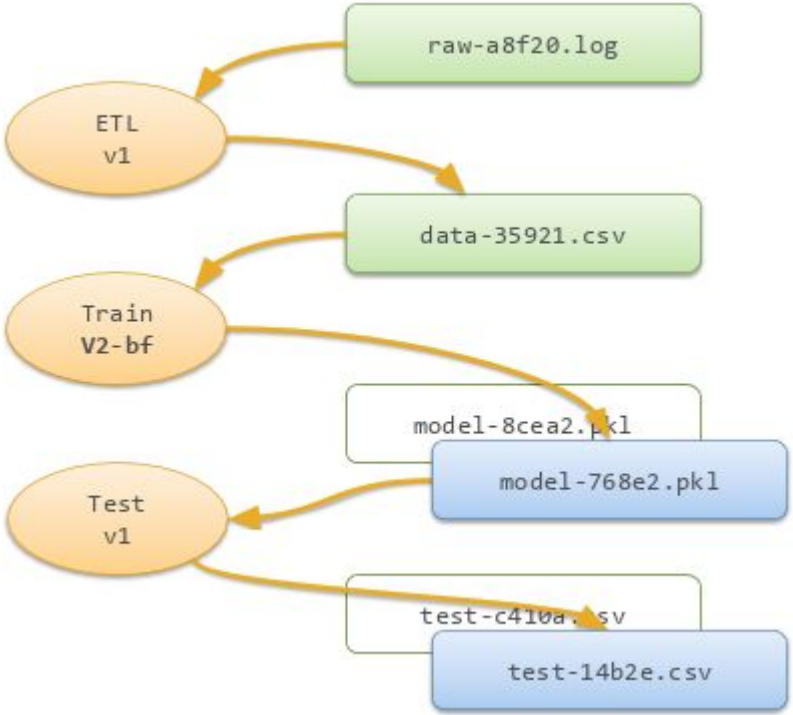


Salted: Git


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: Graphs





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
```

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
?
```

Inheritance vs Composition

```
class Animal():
    def walk():
        ...
    def play():
        ...

class Cat(Animal):
    pass

    def walk():
        ...

cat = Cat()
cat.walk()
cat.play()
...
```

Inheritance vs Composition

```
class Animal():
    def walk():
        ...
    def play():
        ...

class Cat(Animal):
    pass

    def walk():
        ...

cat = Cat()
cat.walk()
cat.play()
...
```

```
class Cat:

    def __init__(self):
        self.animal = Animal()

    def walk(self):
        self.animal.walk()

    def __getattr__(self, attr):
        return getattr(self.animal, attr)
```

Q&A