

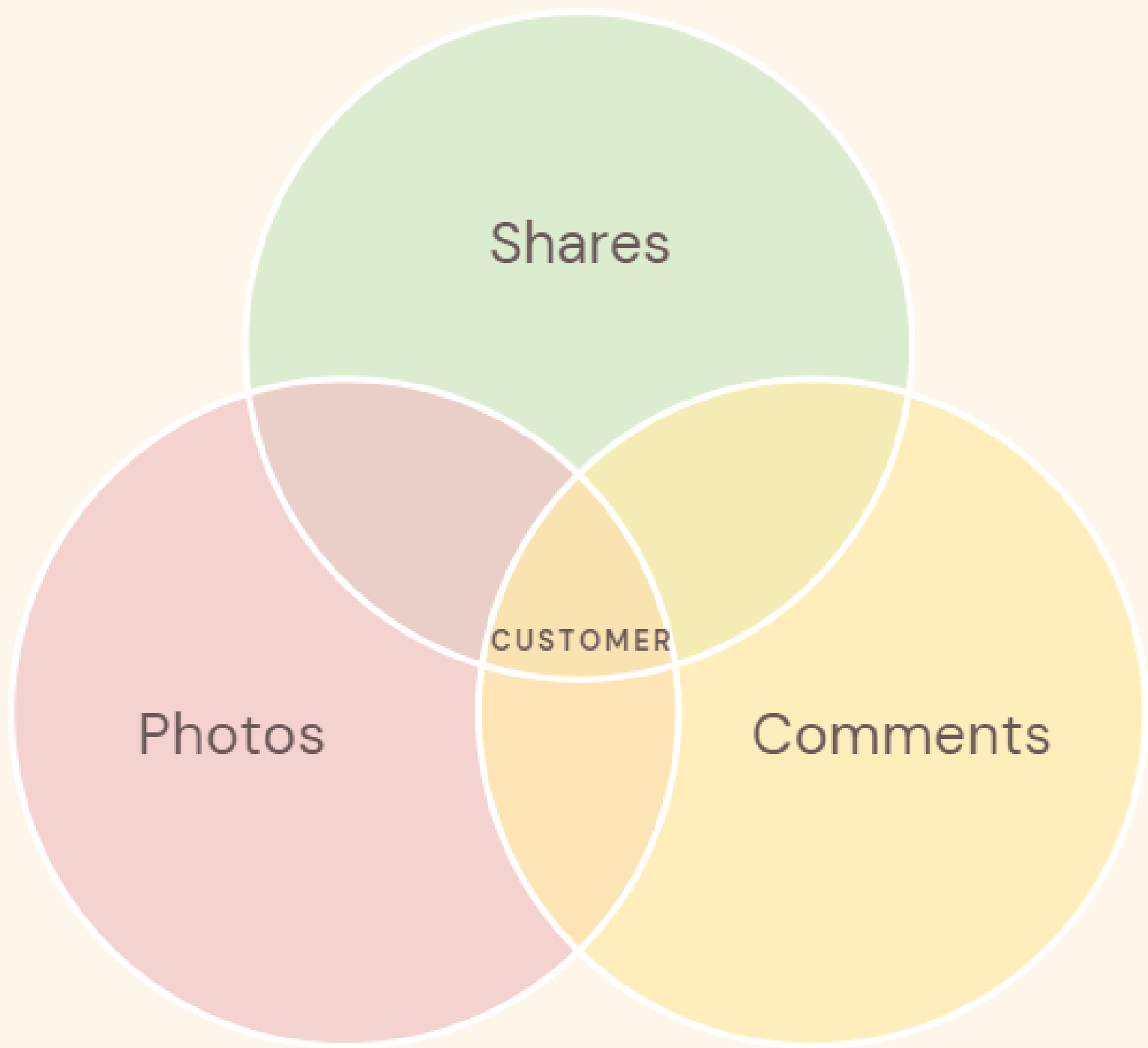


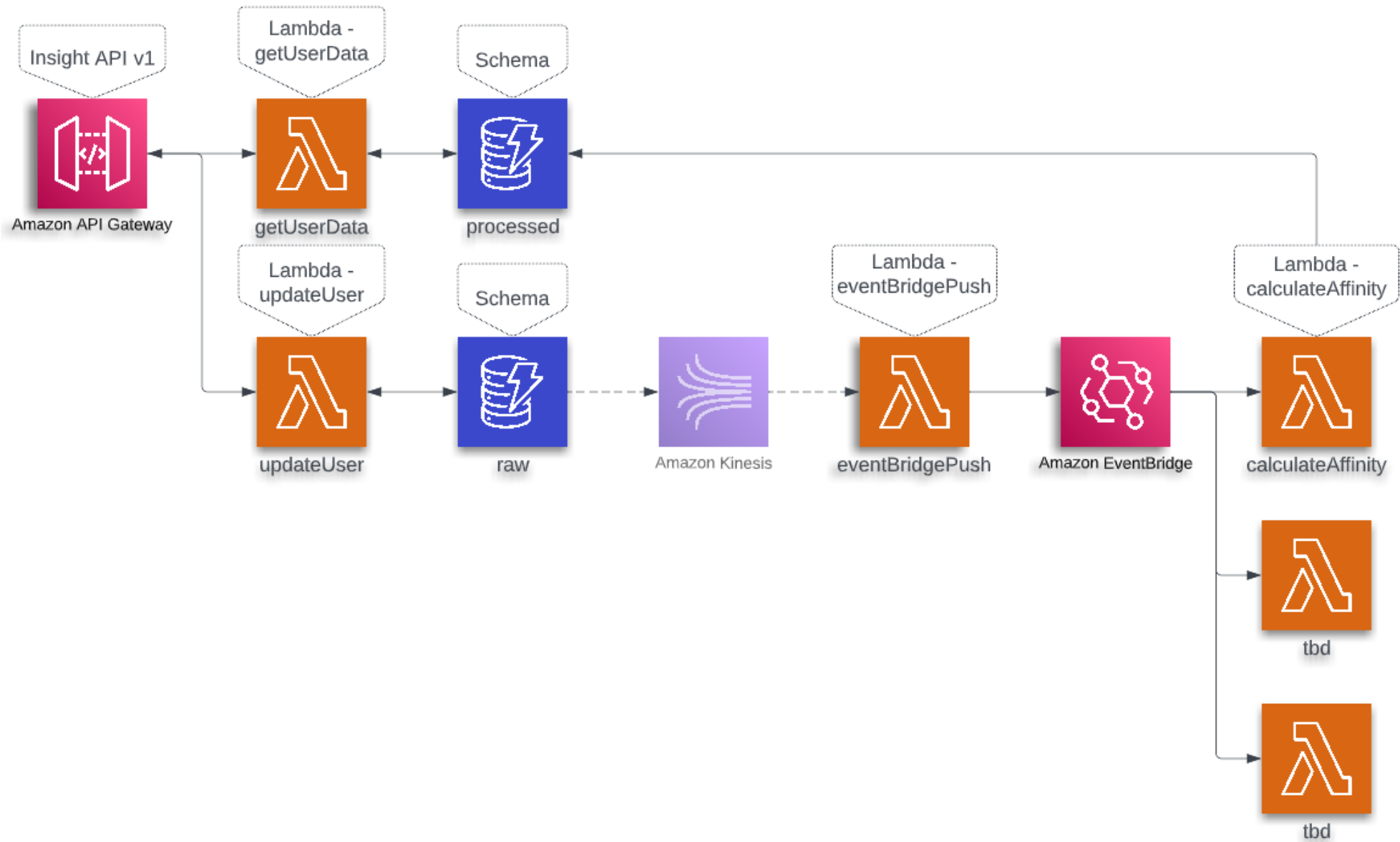
Project Insight

MATTHEW CLEMENCE, JULIO BONILLA, RYAN DRAKE, DYLAN VAUGHN, LOGAN REED

The Problem

- Track user actions
- Determine customer affinity
- Reduce analysis by client

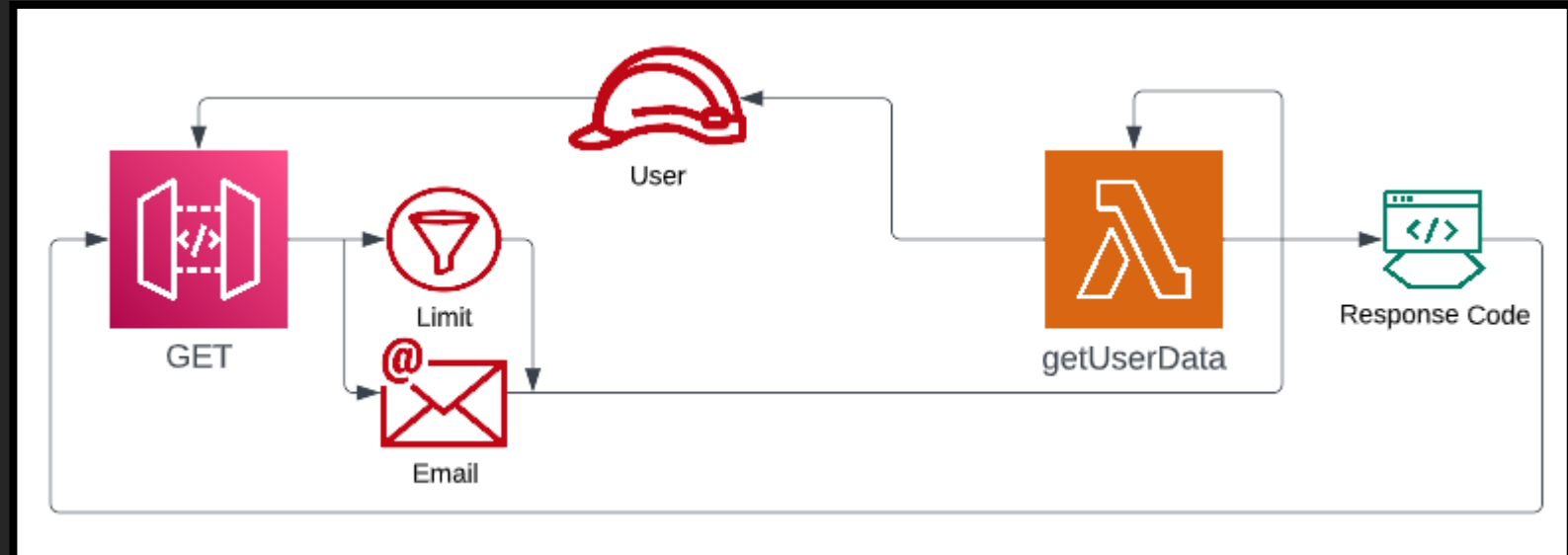




API

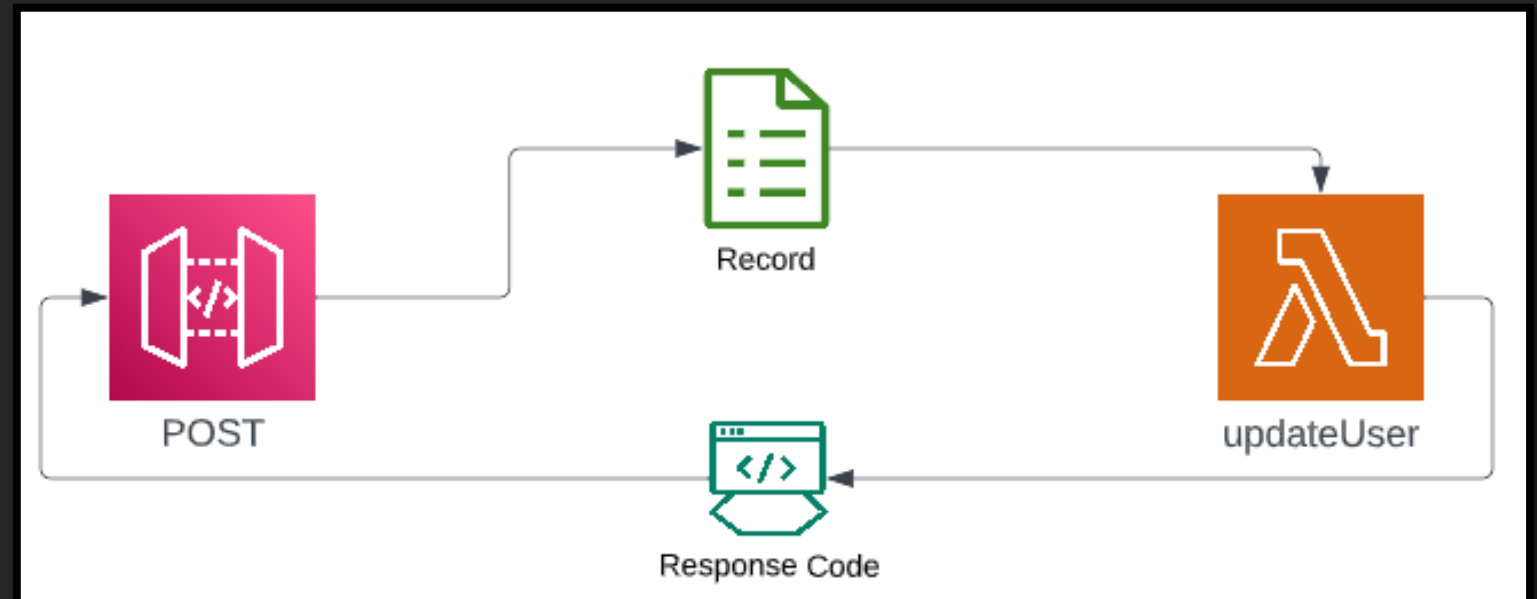
GET method

- Query string parameters
- Returns User object and response code.



POST method

- Record object
- Returns response code.



Lambda - updateUser

- Updates the user's information in the Raw table of the database
- The Raw table of the database has 1 entry per user
- Contains the sum of the raw data passed in from the client
- Multiple "categories" - ex: Photos, Shares, Comments

Lambda - getUserData

- Creates a response payload with the Processed data corresponding to the user
- The Processed table contains a history of entries for each user
- Each entry contains an affinity calculation and the calculated "angle" between each category of data

```
mirror_mod = modifier_ob.modifiers.new("mirror_mod")
# Add mirror object to mirror_ob
mirror_mod.mirror_object = mirror_ob

# MIRROR_X
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
# MIRROR_Y
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
# MIRROR_Z
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

# Selection at the end -add back the deselected
mirror_ob.select = 1
modifier_ob.select = 1
key.context.scene.objects.active = modifier_ob
key.context.selected_objects[0].name = "selected" + str(modifier_ob) # modifier_ob
mirror_ob.select = 0
key.context.selected_objects[0].name = "selected" + str(modifier_ob)
key.context.objects[key.context.selected_objects[0].name].select = 1

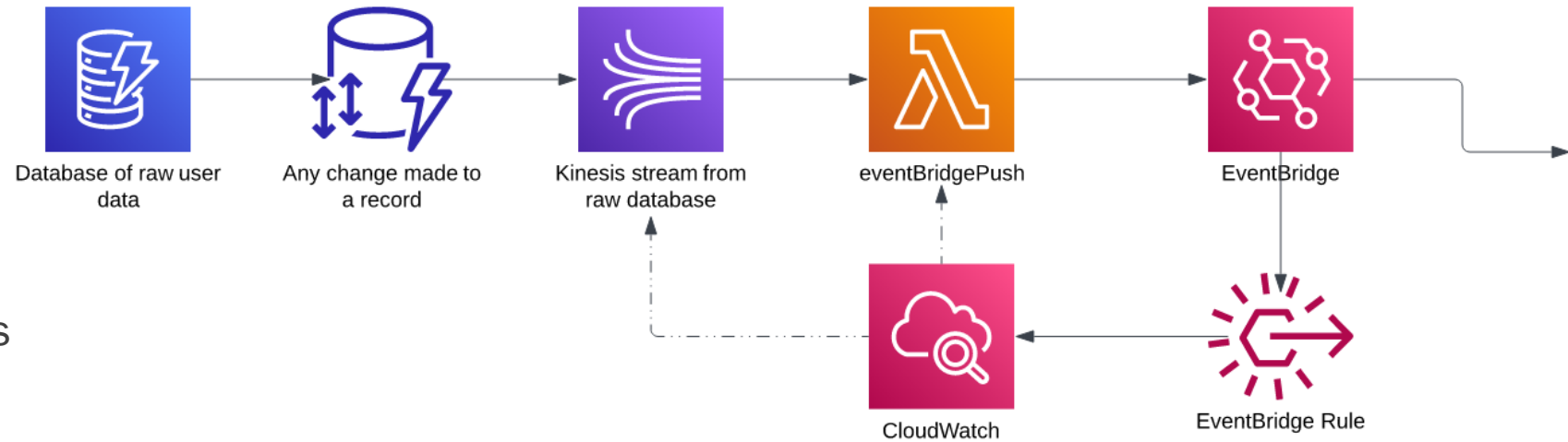
print("please select exactly two objects,")

OPERATOR CLASSES -----
class Operator(bpy.types.Operator):
    """Operator to mirror to the selected object"""
    bl_idname = "mirror_mirror_x"
    bl_label = "Mirror X"

    def execute(self, context):
        if context.active_object is not None
```

Listening for events to trigger the next step, Change Data Capture

- Kinesis data stream triggers Lambda eventBridgePush
- eventBridgePush sends event payload to specified event bus
- EventBridge then invokes calculateAffinity



Lambda - calculateAffinity

- Receives an event payload from EventBridge
- Calculate magnitude and N angles
- Queries the database to create an item in the Processed table for a specific user

Thanks for listening!