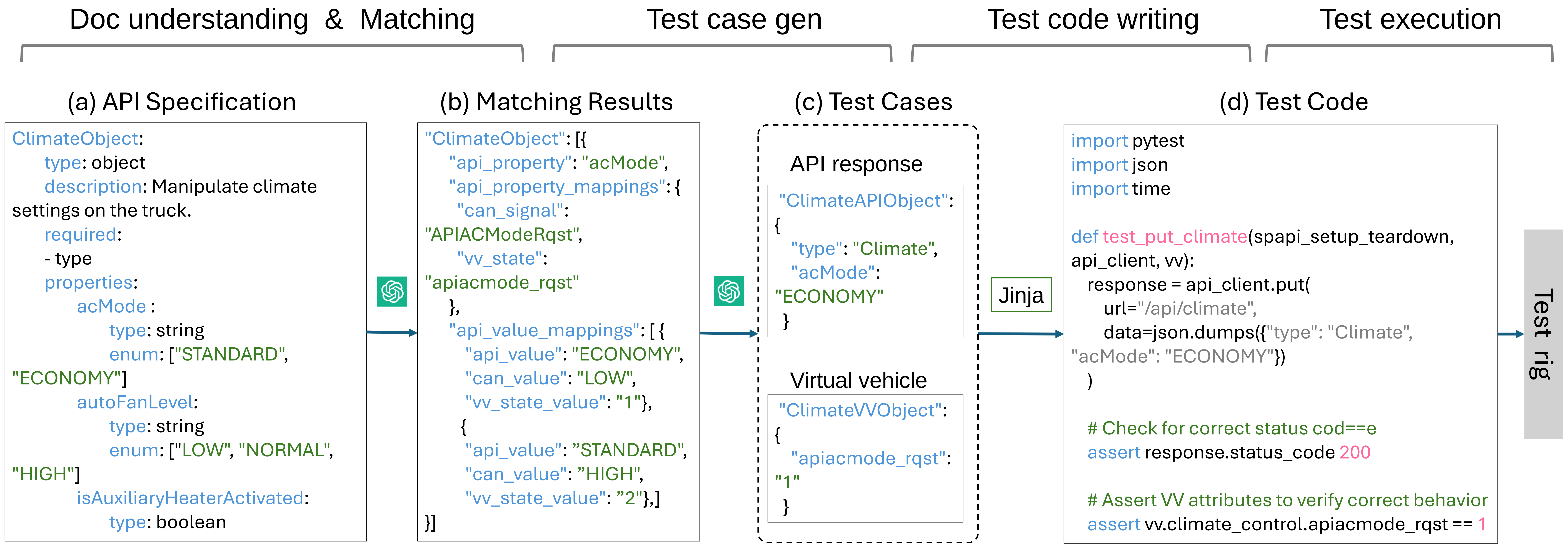

```
ClimateObject:
    type: object
    description: Manipulate climate
settings on the truck.
    required:
    - type
    properties:
        acMode :
            type: string
            enum: ["STANDARD",
"ECONOMY"]
        autoFanLevel:
            type: string
            enum: ["LOW", "NORMAL",
"HIGH"]
        isAuxiliaryHeaterActivated:
            type: boolean
```

```
"ClimateObject": [{
   "api_property": "acMode",
   "api_property_mappings": {
    "can_signal":
"APIACModeRqst",
    "vv_state":
"apiacmode_rqst"
   },
   "api_value_mappings": [ {
     "api_value": "ECONOMY",
     "can_value": "LOW",
     "vv_state_value": "1"},
    {
     "api_value": "STANDARD",
     "can_value": "HIGH",
     "vv_state_value": "2"},]
}]
```

```
 "ClimateAPIObject":
{
   "type": "Climate",
   "acMode":
"ECONOMY"
 }
```

```
 "ClimateVVObject":
{
   "apiacmode_rqst":
"1"
 }
```

```
import pytest
import json
import time

def test_put_climate(spapi_setup_teardown,
api_client, vv):
  response = api_client.put(
    url="/api/climate",
    data=json.dumps({"type": "Climate",
"acMode": "ECONOMY"})
  )

  # Check for correct status cod==e
  assert response.status_code 200

  # Assert VV attributes to verify correct behavior
  assert vv.climate_control.apiacmode_rqst == 1
```