

# IOT Platform

---

Group 5

# Introduction

---

- The rise of Internet of Things (IoT) has enabled us to connect and collect data from an ever-growing number of devices and sensors. However, managing and analyzing this data can be a complex and time-consuming task, which is where our **IoT platform** comes in.
- We have built a **scalable** solution for **deploying** and **managing** large numbers of IoT based applications.

# With our Platform, you can...

---

- Connect to and collect data from plethora of sensors
- Deploy multiple applications
- Compile services to create workflows
- Schedule deployment
- Get authentication through LDAP
- Perform Fault Tolerant and Reliable deployment
- Easily connect to deployed applications and avail it's services
- Also avail other platform services for analyses and notification

# Introduction (Contd...)

---

- Comprises of 7 subsystems

- Bootstrapper
- Application Controller
- Scheduler
- Node Manager
- Load Balancer
- Deployer
- Sensor Manager
- Monitoring & Fault Tolerance

## Techstack

Inter-Module Communication

**Kafka**

Containerization, abstraction

**Docker**

Virtual Machines

**Azure**

DataBase

**MongoDB**

Other File Storage

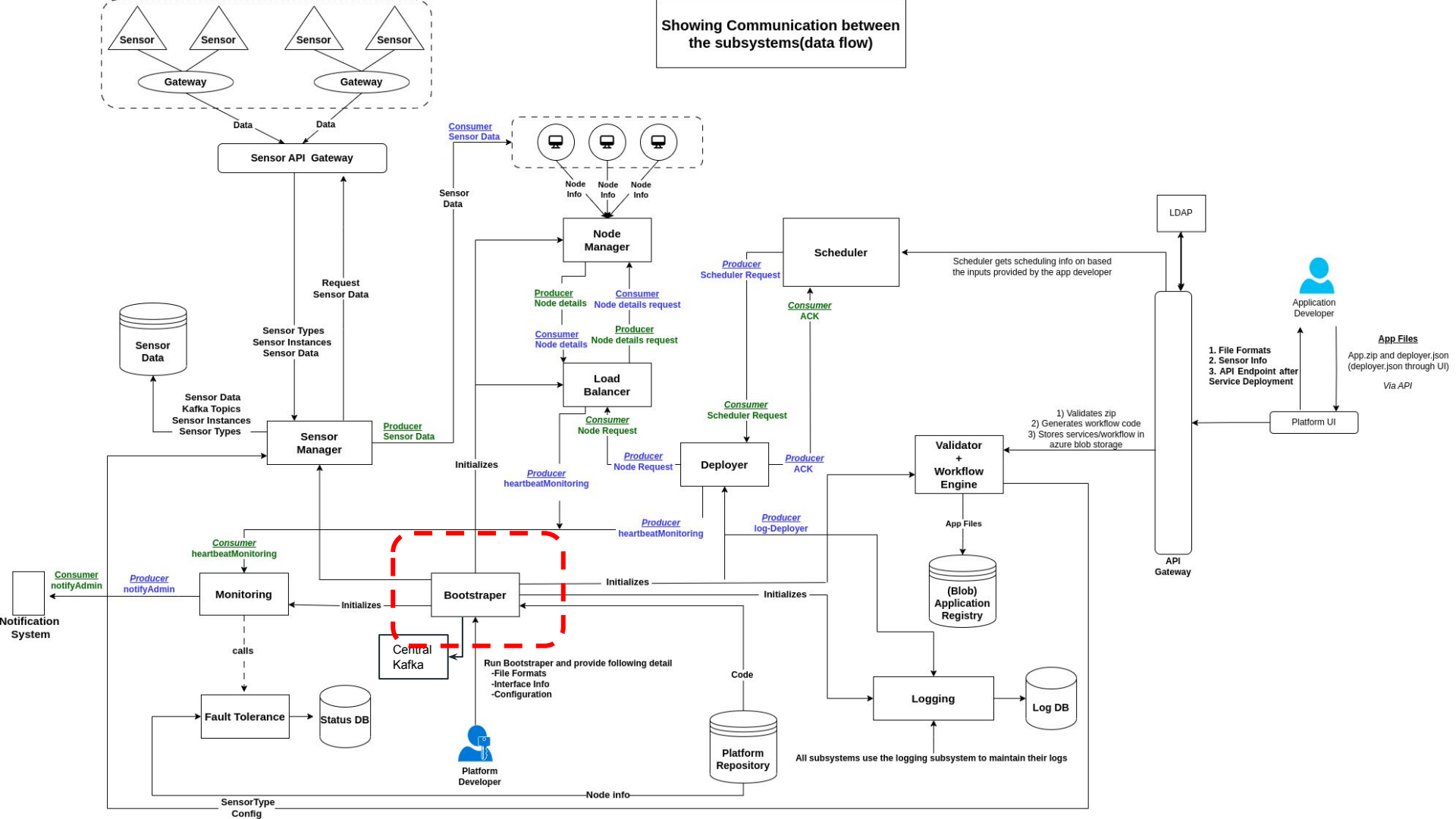
**Azure blob  
storage**

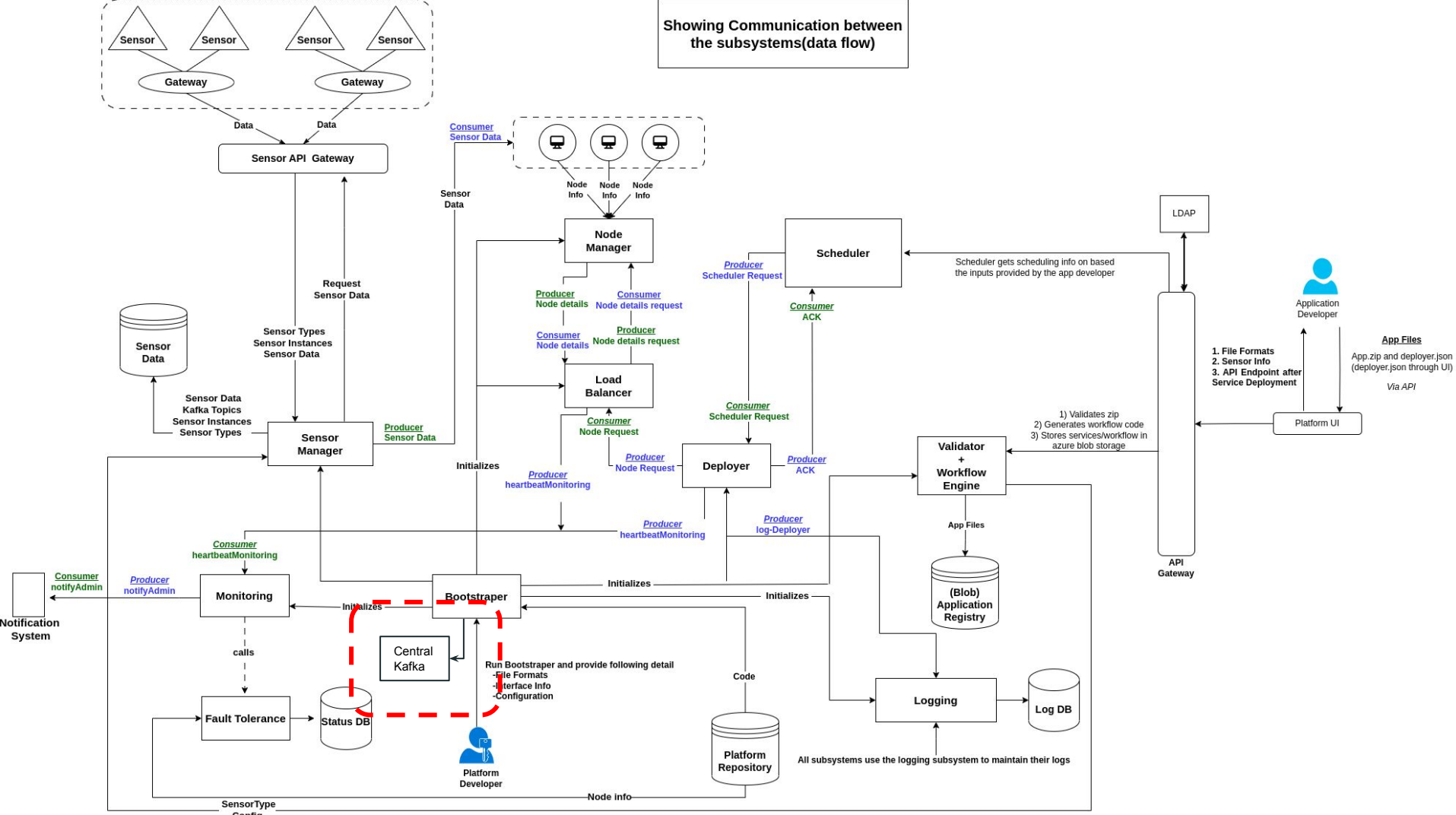
Authentication

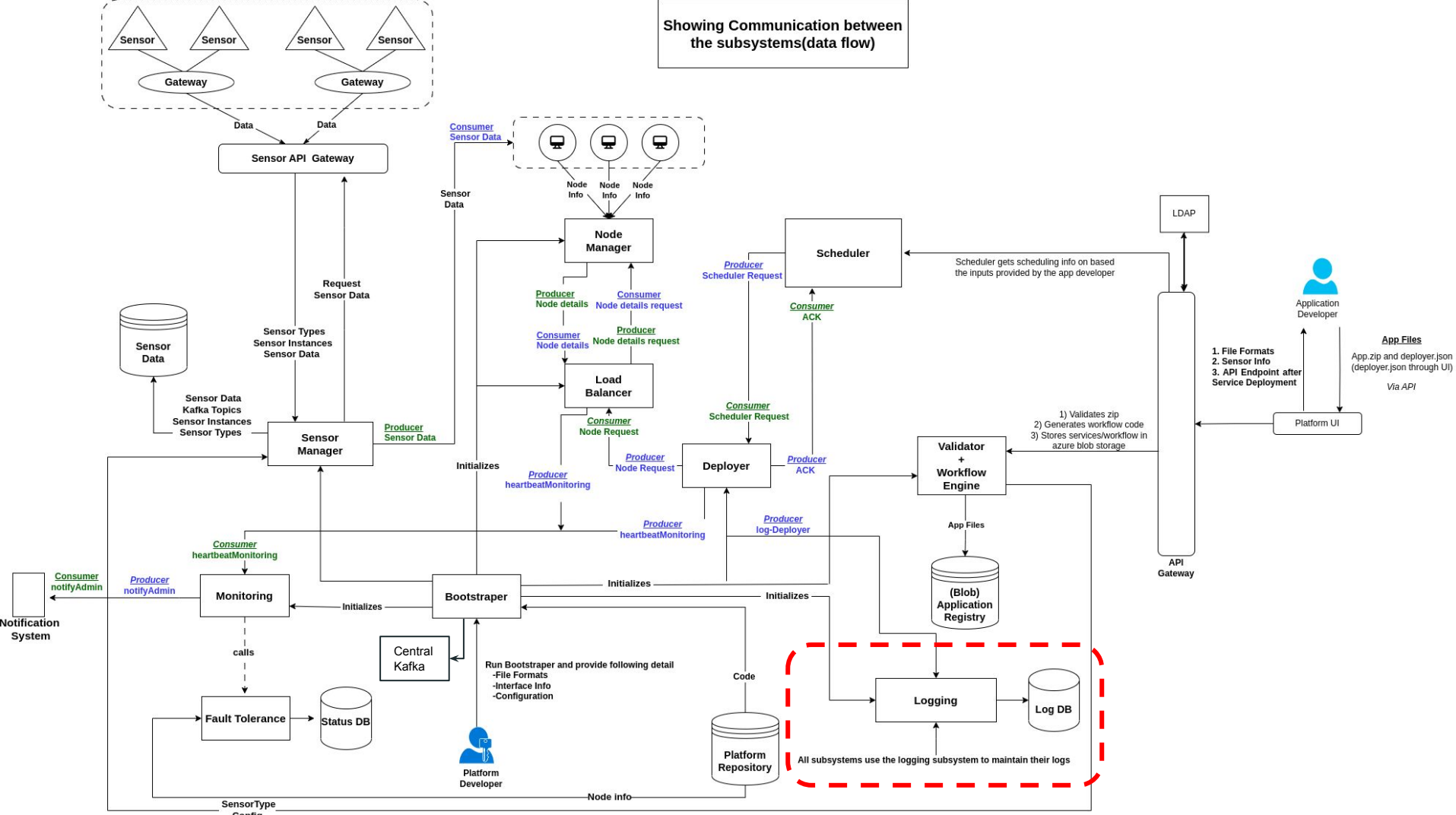
**LDAP**

# Communication Diagram

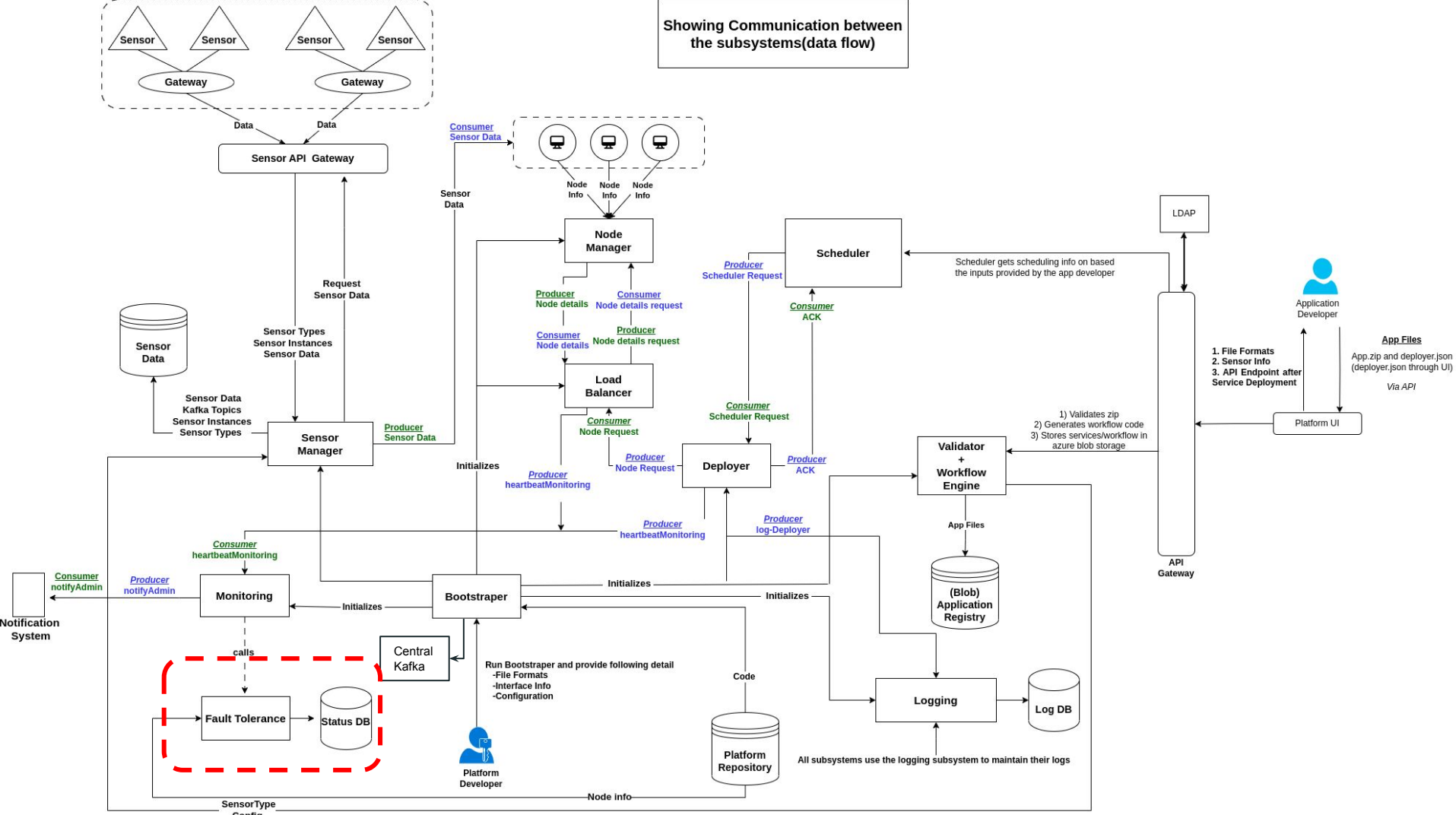
---

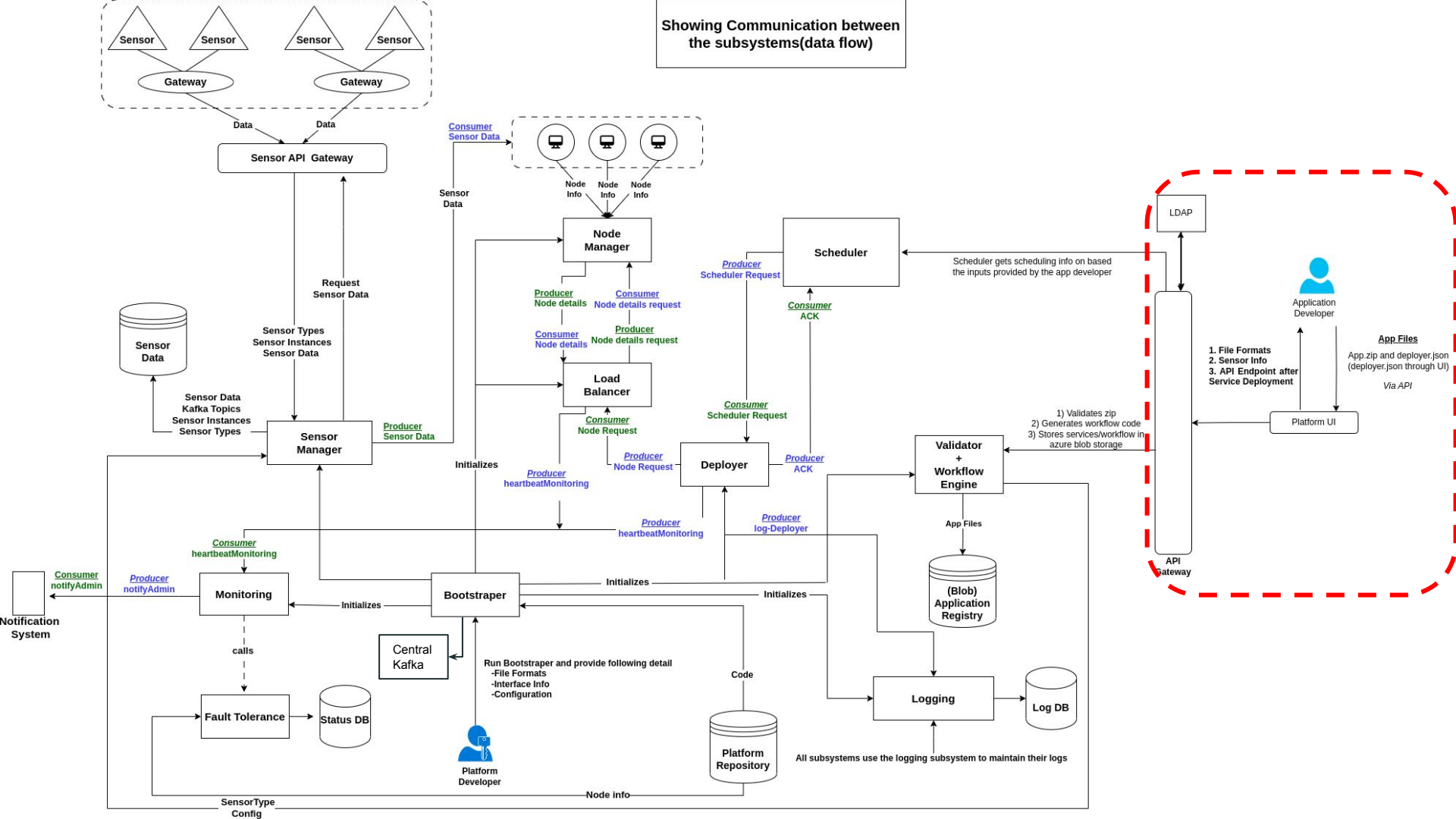


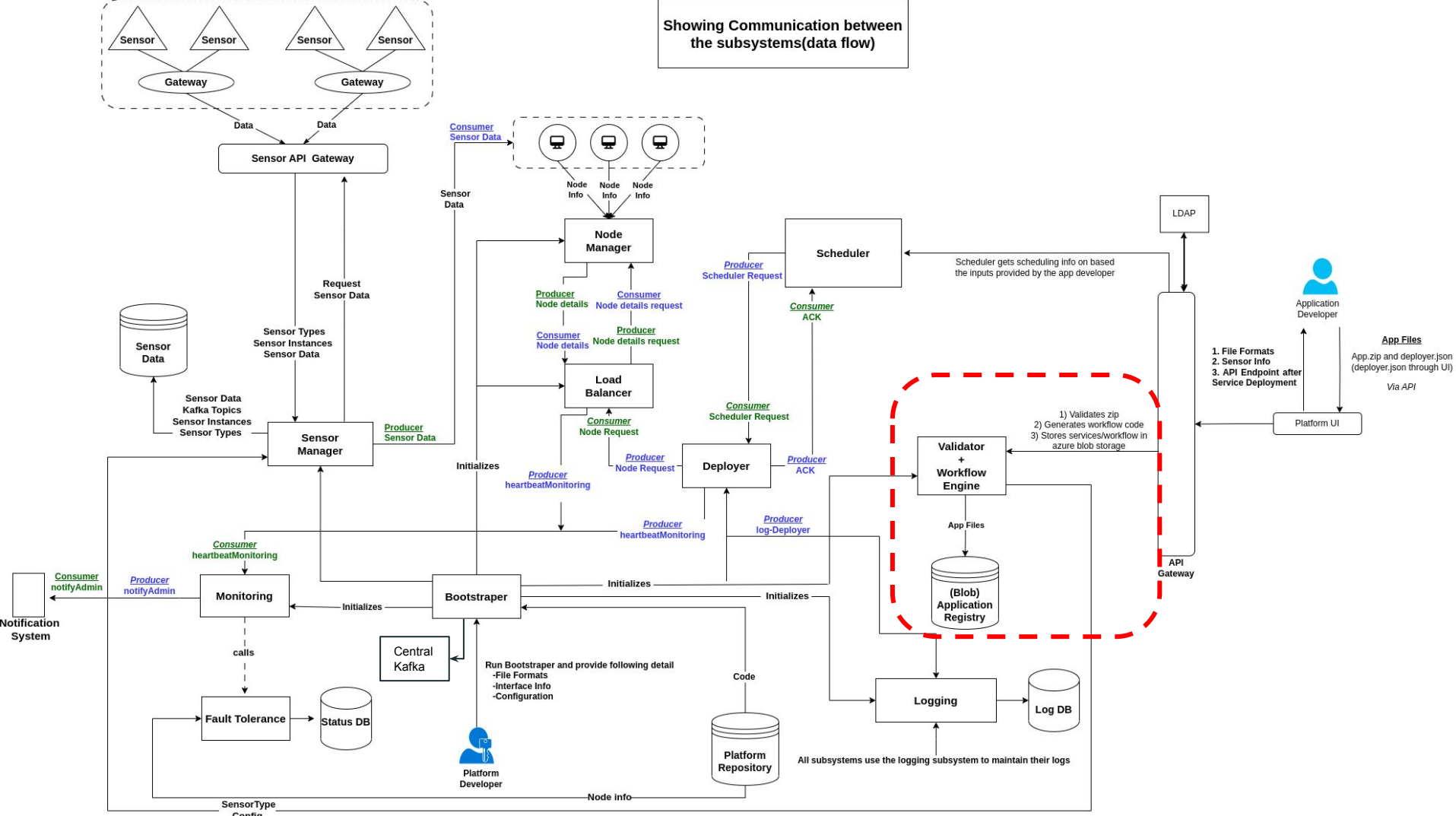


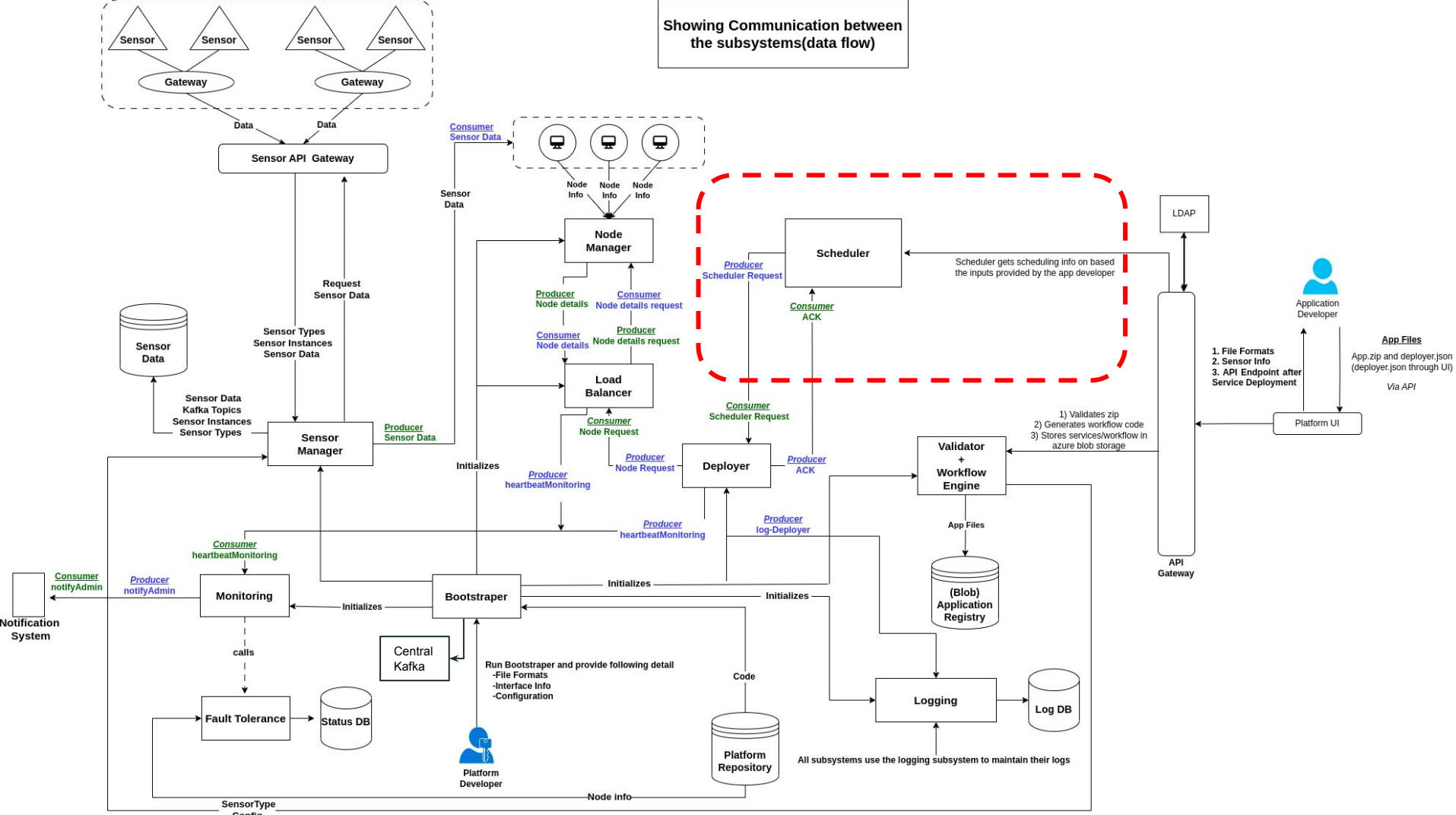


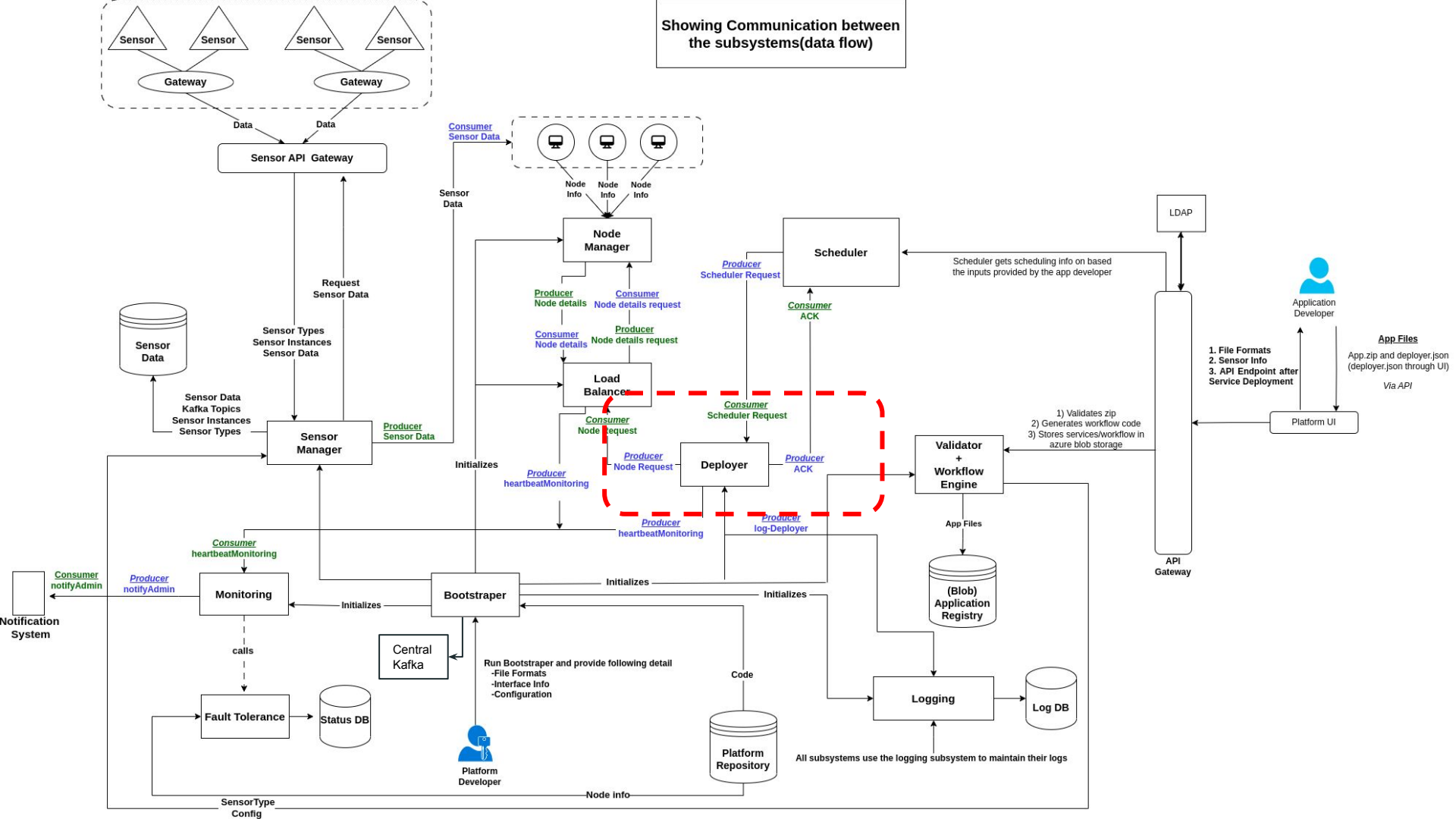


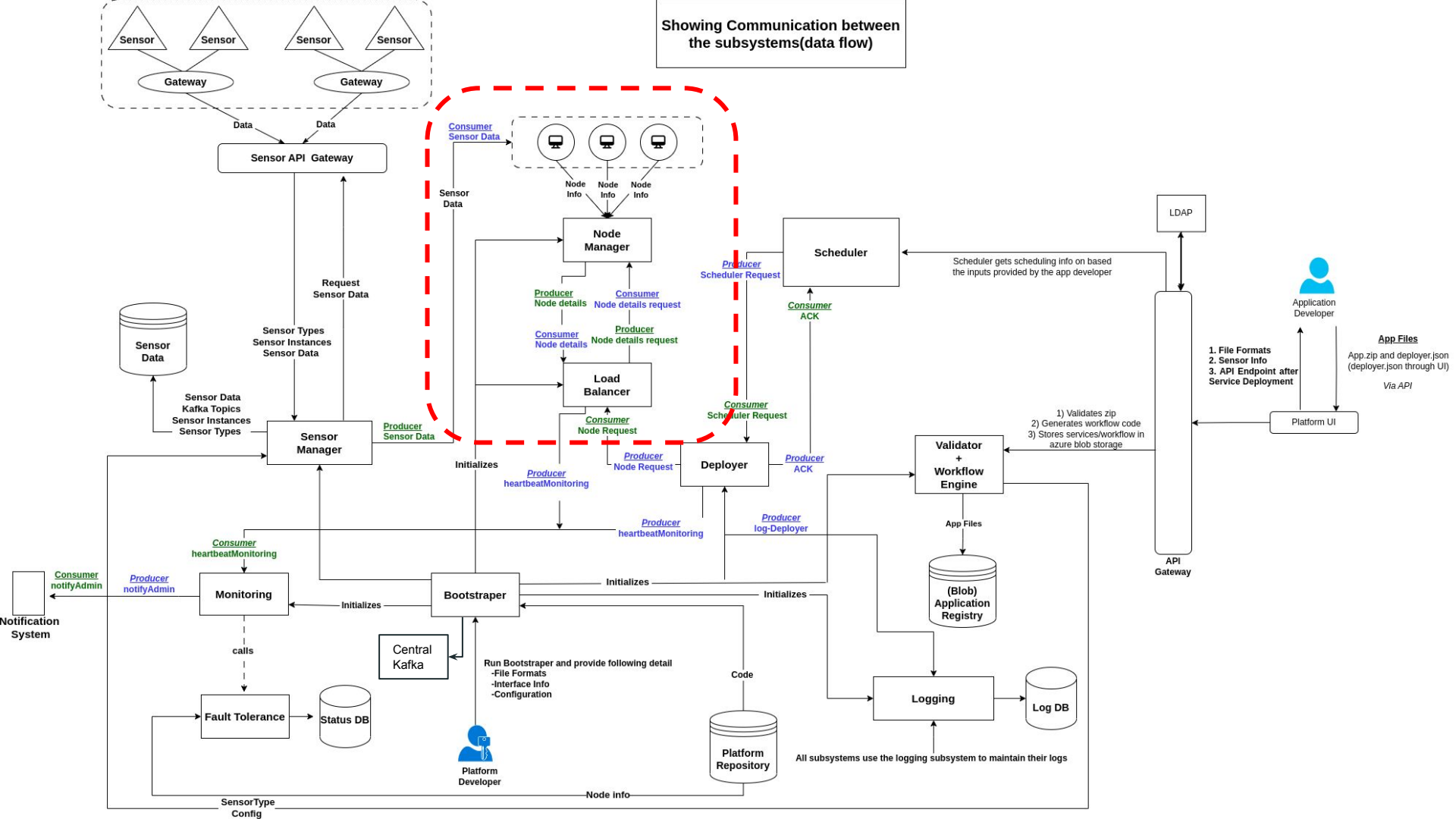


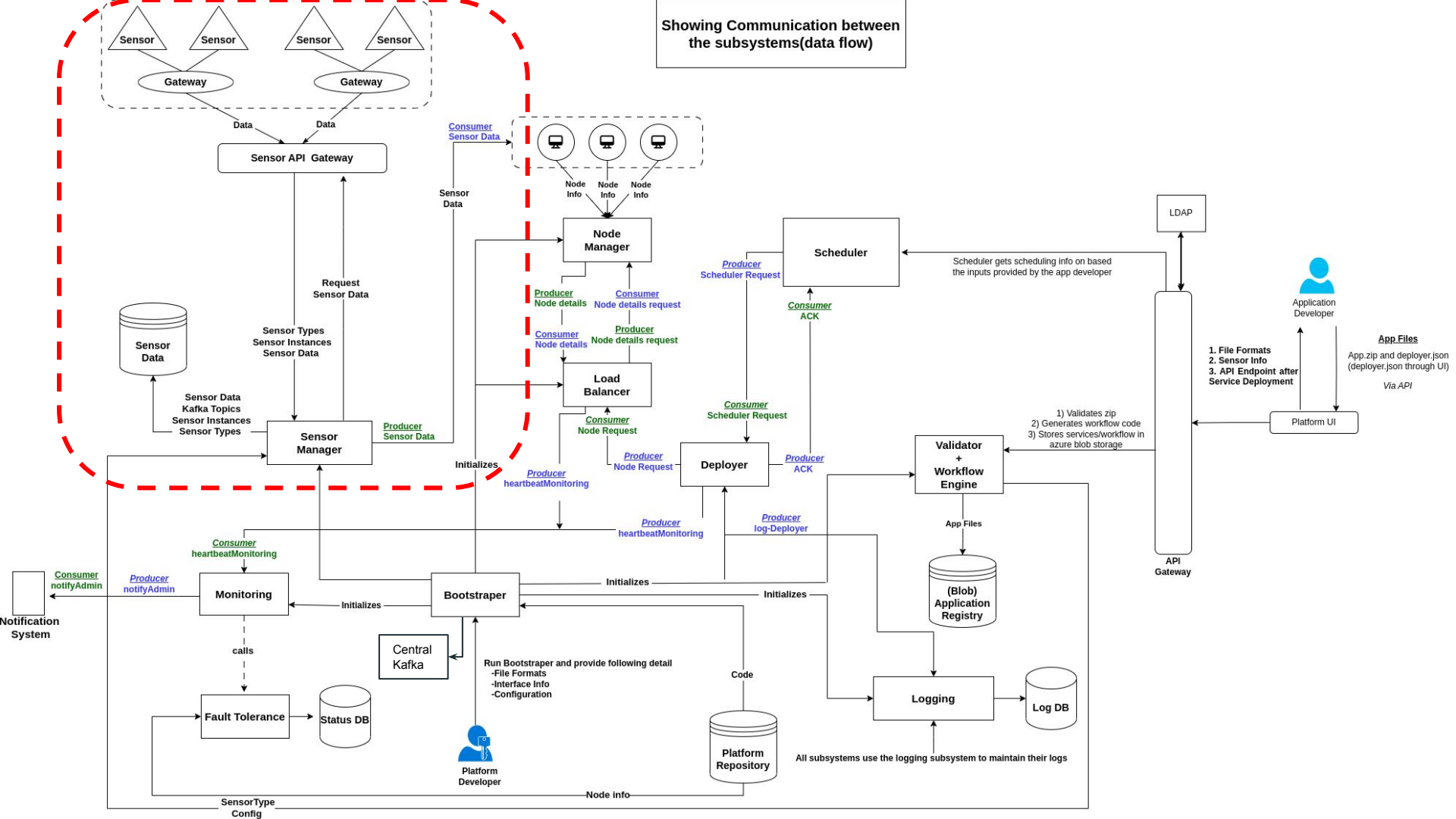












# App Model

---





# Contracts

---

# App Config

```
{
  "applicationName": "finalSampleApp",
  "services": [
    {
      "name": "app",
      "files": [
        "app.py",
        "requirements.txt",
        "config.json"
      ],
      "endpoint": "/server",
      "parameters": [
        {
          "name": "appId",
          "dataType": "str"
        },
        {
          "name": "serviceName",
          "dataType": "str"
        },
        {
          "name": "requestData",
          "dataType": "dict"
        }
      ],
      "sensors": [],
      "outputs": [
        {
          "name": "res",
          "dataType": "dict"
        }
      ]
    }
  ],
}
```



```
{
  "name": "f1",
  "files": [
    "f1.py",
    "requirements.txt",
    "config.json"
  ],
  "endpoint": "/",
  "parameters": [
    {
      "name": "location",
      "dataType": "str"
    },
    {
      "name": "mobile_no",
      "dataType": "str"
    },
    {
      "name": "email",
      "dataType": "str"
    },
    {
      "name": "type",
      "dataType": "str"
    },
    {
      "name": "service_name",
      "dataType": "str"
    },
    {
      "name": "app_id",
      "dataType": "str"
    }
  ],
  "sensors": [
    {
      "sensor_type": "AQ",
      "num_of_sensors": 3
    },
    {
      "sensor_type": "SE",
      "num_of_sensors": 2
    }
  ],
  "outputs": [
    {
      "name": "res1",
      "dataType": "str"
    },
    {
      "name": "res2",
      "dataType": "int"
    }
  ]
}
```

```
{
  "name": "f2",
  "files": [
    "f2.py",
    "requirements.txt",
    "config.json"
  ],
  "endpoint": "/",
  "parameters": [
    {
      "name": "location",
      "dataType": "str"
    },
    {
      "name": "mobile_no",
      "dataType": "str"
    },
    {
      "name": "email",
      "dataType": "str"
    },
    {
      "name": "type",
      "dataType": "str"
    },
    {
      "name": "service_name",
      "dataType": "str"
    },
    {
      "name": "app_id",
      "dataType": "str"
    }
  ],
  "sensors": [
    {
      "sensor_type": "AQ",
      "num_of_sensors": 3
    },
    {
      "sensor_type": "SE",
      "num_of_sensors": 2
    },
    {
      "sensor_type": "REM",
      "num_of_sensors": 1
    }
  ],
  "outputs": [
    {
      "name": "res1",
      "dataType": "str"
    },
    {
      "name": "res2",
      "dataType": "int"
    }
  ]
},
{
  "workflows": [
    "w1.json"
  ],
  "developer_id": ""
}
```

# Workflow

```
{
  "workflowName": "sampleWorkflowInfo",
  "workflowInputs": [
    {
      "name": "appId",
      "dataType": "str",
      "required": true
    },
    {
      "name": "x",
      "dataType": "int"
    },
    {
      "name": "y",
      "dataType": "str"
    }
  ],
  "services": [
    {
      "serviceName": "foo",
      "endpoint": "/foo",
      "parameters": [
        {
          "name": "par_1",
          "dataType": "int",
          "prevOutput": false,
          "prevServiceName": null,
          "prevOutputName": null,
          "workflowInputName": "x"
        }
      ]
    },
    {
      "name": "res1",
      "dataType": "str"
    },
    {
      "name": "res2",
      "dataType": "int"
    }
  ]
}
```

```
Workflow.json
{
  "serviceName": "bar",
  "endpoint": "/bar",
  "parameters": [
    {
      "name": "par_1",
      "dataType": "str",
      "prevOutput": true,
      "prevServiceName": "foo",
      "prevOutputName": "res1",
      "workflowInputName": null
    },
    {
      "name": "par_2",
      "dataType": "int",
      "prevOutput": false,
      "prevServiceName": null,
      "prevOutputName": null,
      "workflowInputName": "y"
    }
  ],
  "outputs": [
    {
      "name": "res1",
      "dataType": "int"
    }
  ],
  "serviceName": "random_rating",
  "endpoint": "/random_rating",
  "parameters": [],
  "outputs": [
    {
      "name": "res1",
      "dataType": "int"
    }
  ]
}
```

```
"workflowOutputs": [
  {
    "serviceName": "foo",
    "serviceParName": "res2",
    "parameterName": "inferene1"
  },
  {
    "serviceName": "bar",
    "serviceParName": "res1",
    "parameterName": "inferene2"
  },
  {
    "serviceName": "random_rating",
    "serviceParName": "res1",
    "parameterName": "inferene3"
  }
]
```

# Thank You!

---

*Any Questions?*