Encryption and Decryption:

```
from cryptography.fernet import Fernet

# Generate a random encryption key
key = Fernet.generate_key()
cipher_suite = Fernet(key)

# Encrypt data
def encrypt(data):
    encrypted_data = cipher_suite.encrypt(data.encode())
    return encrypted_data

# Decrypt data
def decrypt(encrypted_data):
    decrypted_data = cipher_suite.decrypt(encrypted_data)
    return decrypted_data.decode()
```

Socket Communication

Use Python's socket module to establish a socket connection for sending and receiving data.

```
import socket

# Create a socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('localhost', 12345)  # Update with your desired host and port

# Bind the socket to the server address
server_socket.bind(server_address)

# Listen for incoming connections
server_socket.listen(1)

# Accept incoming connections
client_socket, client_address = server_socket.accept()
```

Time Series Database (InfluxDB) :

Use InfluxDB to save and manage time-series data. Install and configure InfluxDB accordingly.

```
from influxdb import InfluxDBClient

# Connect to InfluxDB
influx_client = InfluxDBClient(host='localhost', port=8086,
database='your_database')

# Define a function to save data to InfluxDB
def save_to_influxdb(data):
    json_body = [
        {
            "measurement": "your_measurement",
            "tags": {},
            "time": timestamp,
            "fields": {
                "data": data
            }
        }
    ]
    influx_client.write_points(json_body)
```

Frontend Communication :

Expose an API endpoint or use a WebSocket to communicate with the frontend.

```
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/data', methods=['GET'])

def get_data():

    # Retrieve data from the database

    data = retrieve_data_from_influxdb()  # Implement this function
```

```
# You may need to format the data as needed
    return jsonify({"data": data})
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)  # Update host and port as needed
```

Complete the Backend Logic :

- Accept encrypted data from the socket.
- Decrypt and decode the data.
- Save the data to InfluxDB using the save_to_influxdb function.
- Implement the retrieve_data_from_influxdb function to get data when the frontend requests it.