Thank you for providing this detailed overview of the water consumption monitoring and prediction system. I'll create a diagram to illustrate the flow of data and processes in this system using the Mermaid diagramming language.

```mermaid

Water Consumption Monitoring and Prediction System.download-icon {

cursor: pointer;

transform-origin: center;

}

.download-icon .arrow-part {

transition: transform 0.35s cubic-bezier(0.35, 0.2, 0.14, 0.95);

transform-origin: center;

}

button:has(.download-icon):hover .download-icon .arrow-part, button:has(.download-icon):focus-visible .download-icon .arrow-part {

transform: translateY(-1.5px);

}

#mermaid-diagram-rq{font-family:var(--font-geist-sans);font-size:12px;fill:#000000;}#mermaid-diagram-rq .error-icon{fill:#552222;}#mermaid-diagram-rq .error-text{fill:#552222;stroke:#552222;}#mermaid-diagram-rq .edge-thickness-normal{stroke-width:1px;}#mermaid-diagram-rq .edge-thickness-thick{stroke-width:3.5px;}#mermaid-diagram-rq .edge-pattern-solid{stroke-dasharray:0;}#mermaid-diagram-rq .edge-thickness-invisible{stroke-width:0;fill:none;}#mermaid-diagram-rq .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-diagram-rq .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-diagram-rq .marker{fill:#666;stroke:#666;}#mermaid-diagram-rq .marker.cross{stroke:#666;}#mermaid-diagram-rq svg{font-family:var(--font-geist-sans);font-size:12px;}#mermaid-diagram-rq p{margin:0;}#mermaid-diagram-rq .label{font-family:var(--font-geist-sans);color:#000000;}#mermaid-diagram-rq .cluster-label text{fill:#333;}#mermaid-diagram-rq .cluster-label span{color:#333;}#mermaid-diagram-rq .cluster-label span p{background-color:transparent;}#mermaid-diagram-rq .label text,#mermaid-diagram-rq span{fill:#000000;color:#000000;}#mermaid-diagram-rq .node rect,#mermaid-diagram-rq .node circle,#mermaid-diagram-rq .node ellipse,#mermaid-diagram-rq .node polygon,#mermaid-diagram-rq .node path{fill:#eee;stroke:#999;stroke-width:1px;}#mermaid-diagram-rq .rough-node .label text,#mermaid-diagram-rq .node .label text{text-anchor:middle;}#mermaid-diagram-rq .node .katex path{fill:#000;stroke:#000;stroke-width:1px;}#mermaid-diagram-rq .node .label{text-align:center;}#mermaid-diagram-rq .node.clickable{cursor:pointer;}#mermaid-diagram-rq .arrowheadPath{fill:#333333;}#mermaid-diagram-rq .edgePath .path{stroke:#666;stroke-width:2.0px;}#mermaid-diagram-rq .flowchart-link{stroke:#666;fill:none;}#mermaid-diagram-rq .edgeLabel{background-color:white;text-align:center;}#mermaid-diagram-rq .edgeLabel p{background-color:white;}#mermaid-diagram-rq .edgeLabel rect{opacity:0.5;background-color:white;fill:white;}#mermaid-diagram-rq .labelBkg{background-color:rgba(255, 255, 255, 0.5);}#mermaid-diagram-rq .cluster rect{fill:hsl(0, 0%, 98.9215686275%);stroke:#707070;stroke-width:1px;}#mermaid-diagram-rq .cluster text{fill:#333;}#mermaid-diagram-rq .cluster span{color:#333;}#mermaid-diagram-rq div.mermaidTooltip{position:absolute;text-align:center;max-width:200px;padding:2px;font-family:var(--font-geist-sans);font-size:12px;background:hsl(-160, 0%, 93.3333333333%);border:1px solid #707070;border-radius:2px;pointer-events:none;z-index:100;}#mermaid-diagram-rq .flowchartTitleText{text-anchor:middle;font-size:18px;fill:#000000;}#mermaid-diagram-rq .flowchart-link{stroke:rgb(var(--gray-400));stroke-width:1px;}#mermaid-diagram-rq .marker,#mermaid-diagram-rq marker,#mermaid-diagram-rq marker \*{fill:rgb(var(--gray-400))!important;stroke:rgb(var(--gray-400))!important;}#mermaid-diagram-rq .label,#mermaid-diagram-rq text,#mermaid-diagram-rq text>tspan{fill:rgb(var(--black))!important;color:rgb(var(--black))!important;}#mermaid-diagram-rq .background,#mermaid-diagram-rq rect.relationshipLabelBox{fill:rgb(var(--white))!important;}#mermaid-diagram-rq .entityBox,#mermaid-diagram-rq .attributeBoxEven{fill:rgb(var(--gray-150))!important;}#mermaid-diagram-rq .attributeBoxOdd{fill:rgb(var(--white))!important;}#mermaid-diagram-rq .label-container,#mermaid-diagram-rq rect.actor{fill:rgb(var(--white))!important;stroke:rgb(var(--gray-400))!important;}#mermaid-diagram-rq line{stroke:rgb(var(--gray-400))!important;}#mermaid-diagram-rq :root{--mermaid-font-family:var(--font-geist-sans);}Monitoring and EnforcementAI/ML and BlockchainData Collection and ProcessingInitial DatasetReal-time DataReal-time DataPredictionsPredictionsConsumption RecordsConsumption DataVisualizationsAlerts/RestrictionsData GenerationIoT SensorsTime-Series ForecastingBlockchainData VisualizationSmart Contract Enforcement

```

This diagram illustrates the flow of data and processes in the water consumption monitoring and prediction system. Here's a breakdown of each component:

1. \*\*Data Generation (A)\*\*: This is the initial step where the dataset is generated, including water consumption data, number of residents, and unit sizes.

2. \*\*IoT Sensors (B)\*\*: These collect real-time water usage data from each unit.

3. \*\*Time-Series Forecasting (C)\*\*: This component uses AI/ML models (like ARIMA, SARIMA, or LSTM) to predict future water consumption based on historical and real-time data.

4. \*\*Blockchain (D)\*\*: This stores water consumption records in a decentralized ledger, ensuring data integrity and transparency.

5. \*\*Data Visualization (E)\*\*: This component creates dashboards and visualizations of water usage data and predictions.

6. \*\*Smart Contract Enforcement (F)\*\*: This final stage enforces usage limits and triggers alerts based on predictions and actual consumption.

The diagram is divided into three main sections:

- \*\*Data Collection and Processing\*\*: This includes the initial data generation and ongoing data collection from IoT sensors.

- \*\*AI/ML and Blockchain\*\*: This section covers the predictive modeling and secure data storage aspects of the system.

- \*\*Monitoring and Enforcement\*\*: This includes the visualization of data and the enforcement of usage limits through smart contracts.

The arrows show how data flows between these components, creating a comprehensive system for monitoring, predicting, and managing water consumption in a multi-unit building.

This diagram provides a high-level overview of the system. If you'd like more detail on any specific part of the system or a different type of visualization, please let me know.

====================================================================================================================================================================

I'll generate code for different components of this water consumption monitoring and prediction system using Python and Solidity. I'll break it down into several parts to cover the main aspects of the project.

Let's start with the data generation and IoT simulation in Python:

```python project="Water Consumption Monitoring" file="data\_generation.py"

...

```

Now, let's create a Python script for time-series forecasting using SARIMA:

```python project="Water Consumption Monitoring" file="time\_series\_forecasting.py"

...

```

Next, let's create a Solidity smart contract for managing water consumption records and enforcing usage limits:

```plaintext project="Water Consumption Monitoring" file="WaterConsumptionContract.sol"

...

```

Finally, let's create a Python script for data visualization:

```python project="Water Consumption Monitoring" file="data\_visualization.py"

...

```

These code snippets cover the main components of the water consumption monitoring and prediction system:

1. Data generation and IoT sensor simulation

2. Time-series forecasting using SARIMA

3. Blockchain implementation with a Solidity smart contract

4. Data visualization

To implement the full system, you would need to:

1. Set up a blockchain network (e.g., using Ethereum or Hyperledger Fabric) and deploy the smart contract.

2. Create a backend service to handle the IoT data stream, interact with the blockchain, and run the forecasting models.

3. Develop a frontend application to display the visualizations and allow users to interact with the system.

4. Implement proper error handling, data validation, and security measures throughout the system.

Remember that this is a simplified version of the system and would need further development and integration to be production-ready. You may also want to consider using more advanced machine learning models (like LSTM) for better predictions, especially when dealing with long-term patterns and multiple variables.