



# Advanced Architectures and Control Concepts for MORE MICROGRIDS Work Package B



## Demonstration of MGCC in Laboratory with centralized and decentralized control

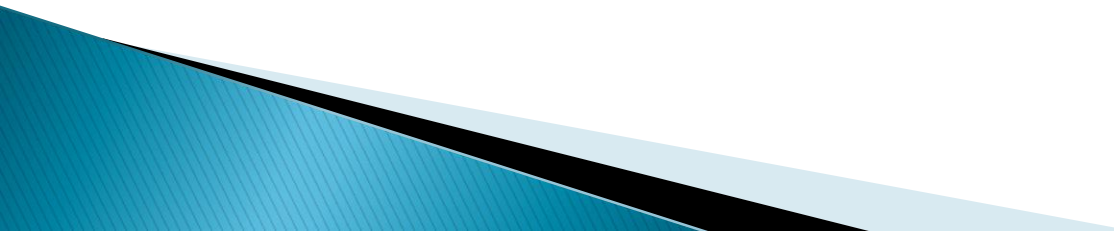
Deliverable DB4 (.ppt)

Access: PUBLIC

Status: Final Version

Authors: Aris Dimeas  
Jose Oyarzabal  
Nikos Hatziargyriou

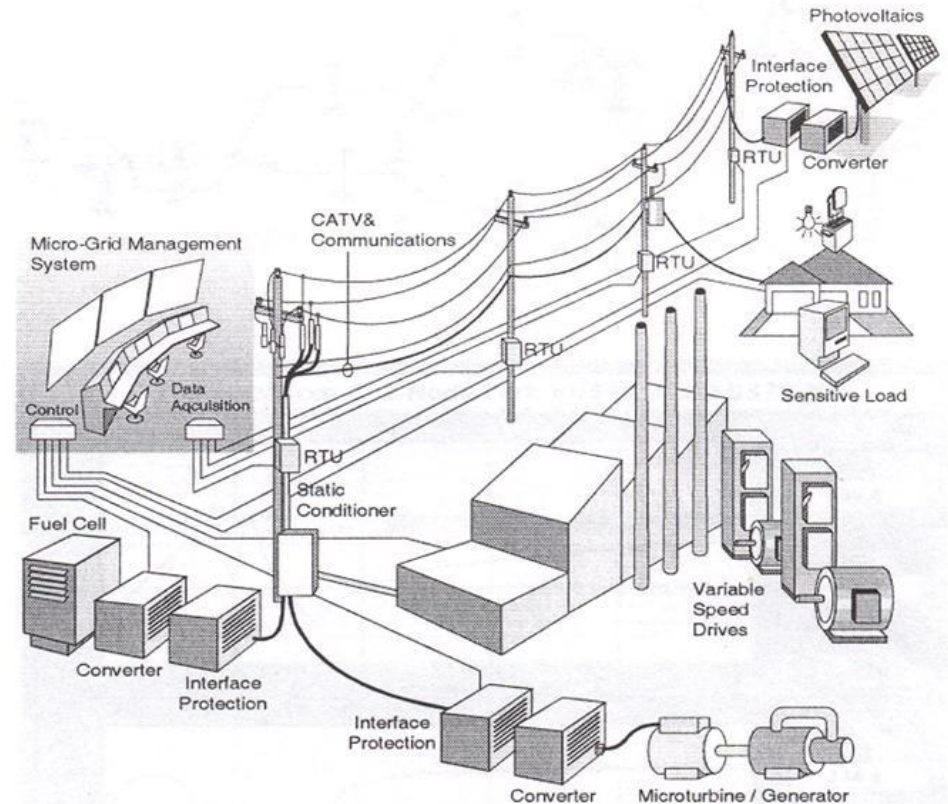
# Outline

- ▶ Introduction
  - ▶ Advanced architectures and control concepts for Microgrids
  - ▶ Development of alternative control strategies
    - centralized vs. decentralized
  - ▶ Examples of implementation
- 

# What are MICROGRIDS?

Interconnection of small, modular **generation to low voltage distribution systems** forms a new type of power system, the Microgrid.

Microgrids can be **connected to the main power network** or be **operated islanded**, in a coordinated, controlled way.



# Why MICROGRIDS?

## Economy

- Production near Load
- No Transmission Cost
- DSM Policies

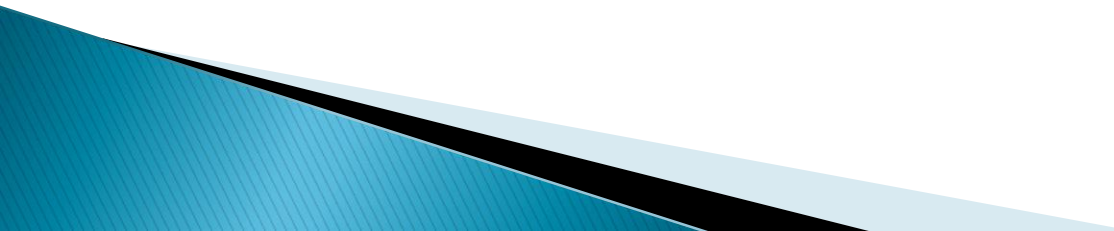
## Reliability

- Power Quality
- Better Reliability indices
- Advanced Capabilities (island mode)


## Environment

- Increased Penetration of RES
- CO2 Reduction
- Energy Efficiency

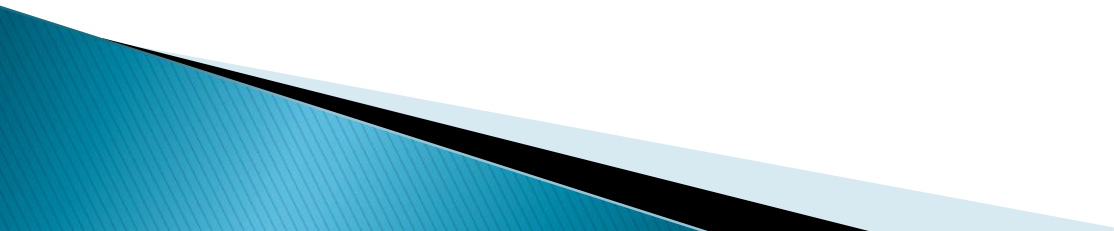
# Technical Challenges

- ▶ Control & Coordination of DGs and Loads
  - ▶ More Efficient and Economic DGs: Batteries, Fuel Cell, Flywheels, etc
  - ▶ Power Electronics Interfaces
  - ▶ Protection Scheme / Static Switch
  - ▶ Communication Requirements
  - ▶ Standardization
- 

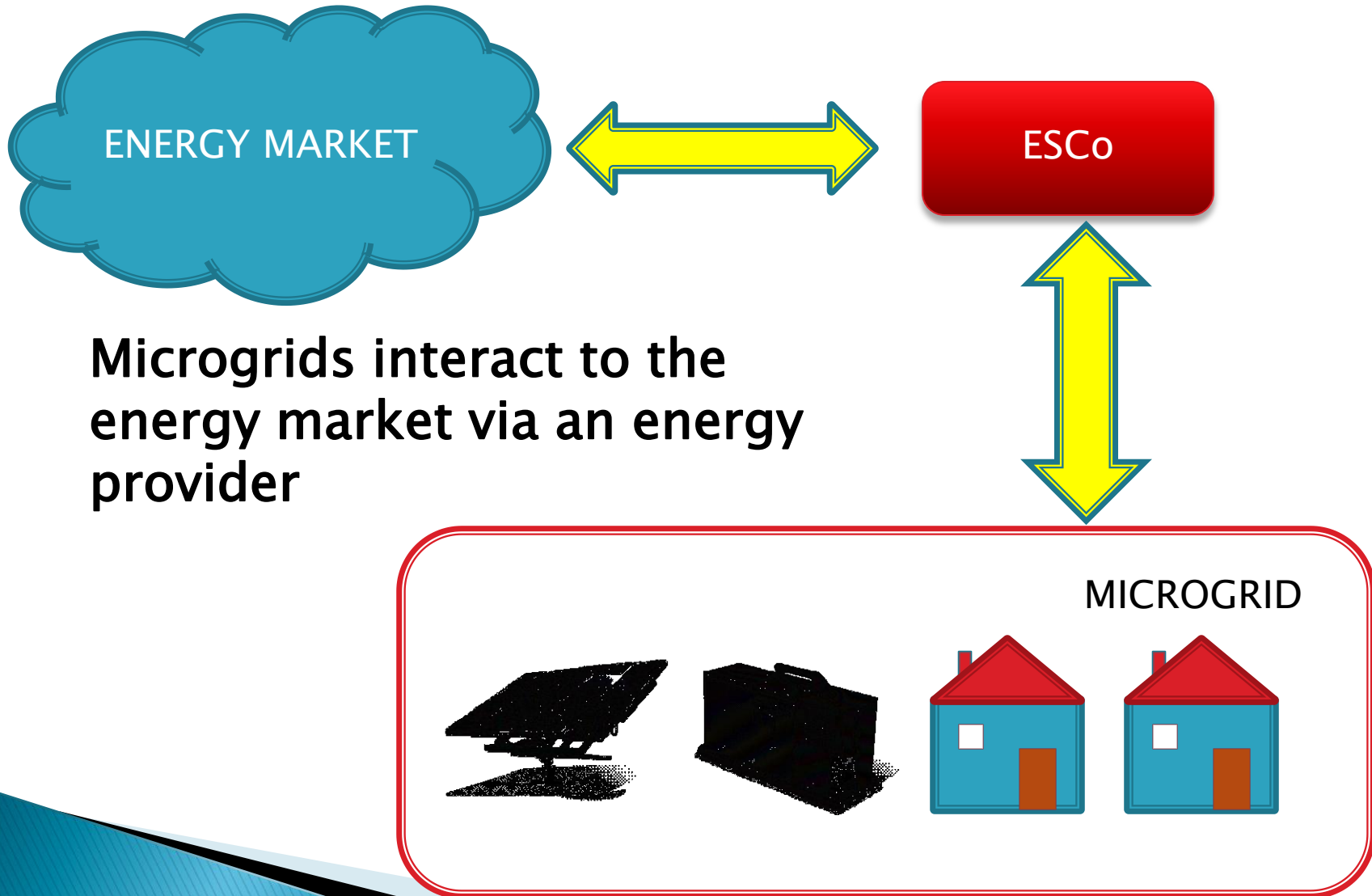
# Control & Coordination: Is it necessary?

- ▶ Why suggest that the coordinated operation of several DGs and Loads (Consumers) may increase the efficiency and will provide useful tools to the network management.
  - ▶ Thus the consumers, the DGs owner and the network may have financial and operational benefits.
  - ▶ These benefits may derive by applying DSM policies, Congestion Management, Black Start, Postpone Transmission/Distribution Network Investments, etc.
- 

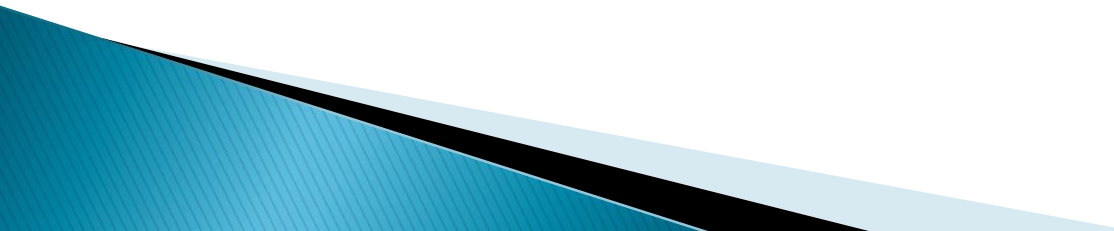
# Basic Challenges for Microgrids Control

- ▶ The legal framework/Market Structure
  - ▶ Larger Number of nodes.
  - ▶ The solution should have very low cost per node.
  - ▶ The Microgrid includes all the operations of large systems.
  - ▶ The system should include DGs from different vendors and different principle of operation.
  - ▶ The available communication infrastructure should be used in order to reduce the cost.
- 

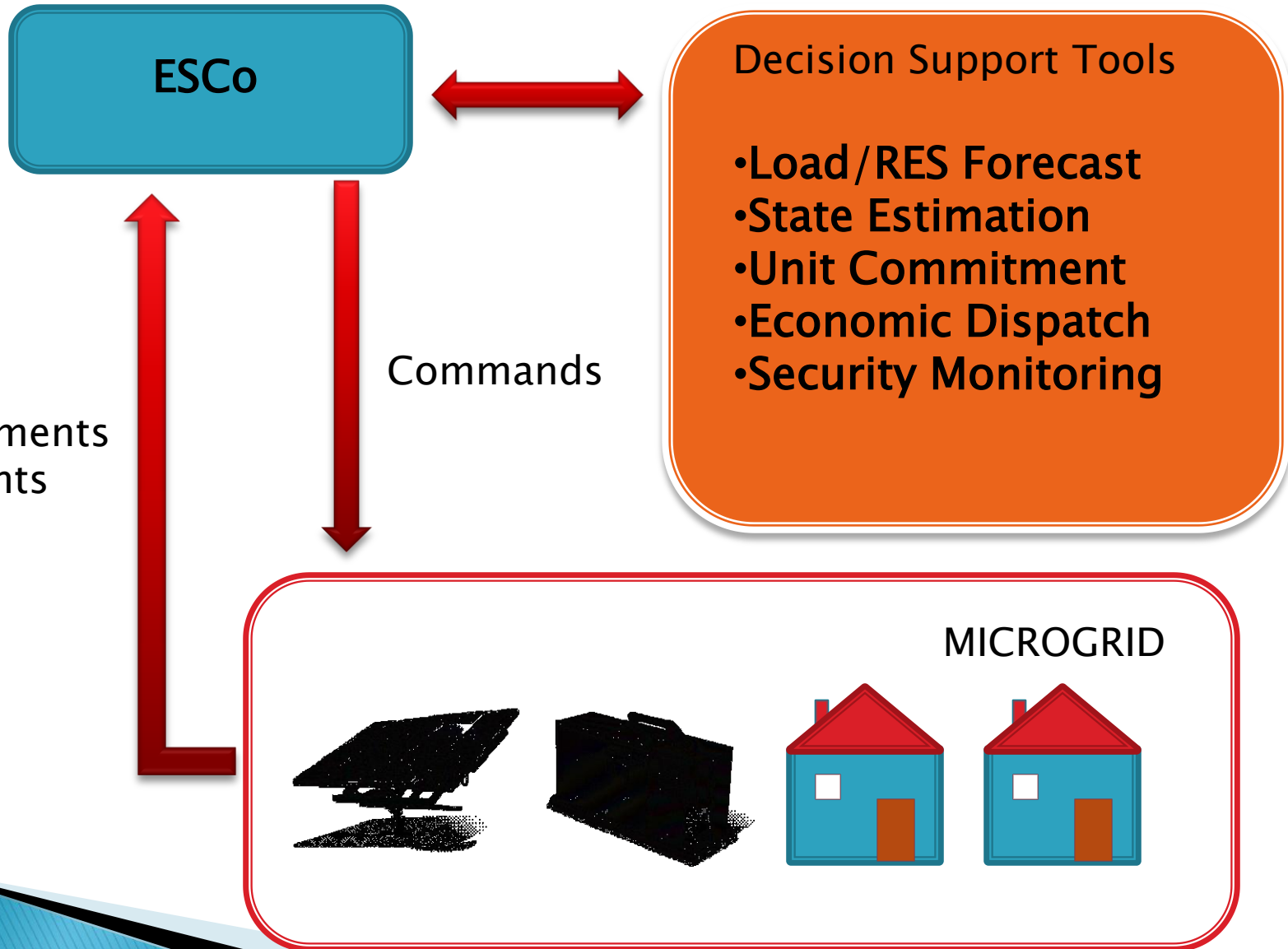
# The Market Structure



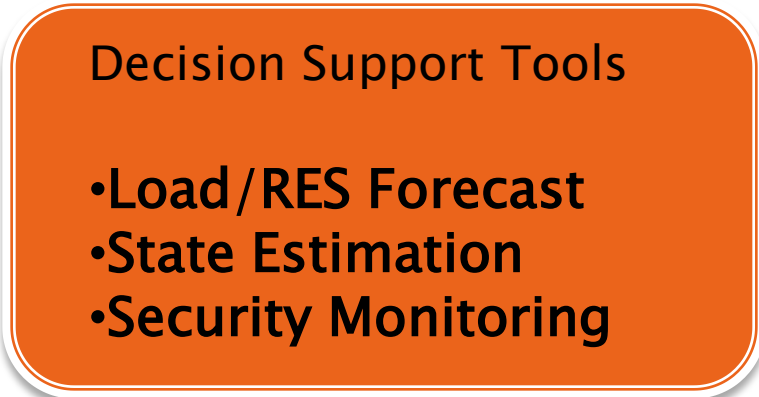
# Centralized & Decentralized Control

- ▶ Two possible control architectures may exist.
  - ▶ The main issue is where the decision is taken
  - ▶ The **Centralized Approach** suggests that a Central Processing Unit collects all the measurement and decides next actions.
  - ▶ The **Decentralized Approach** suggests that advanced controllers are installed in each node forming an distributed control system.
- 

# The Centralized Approach

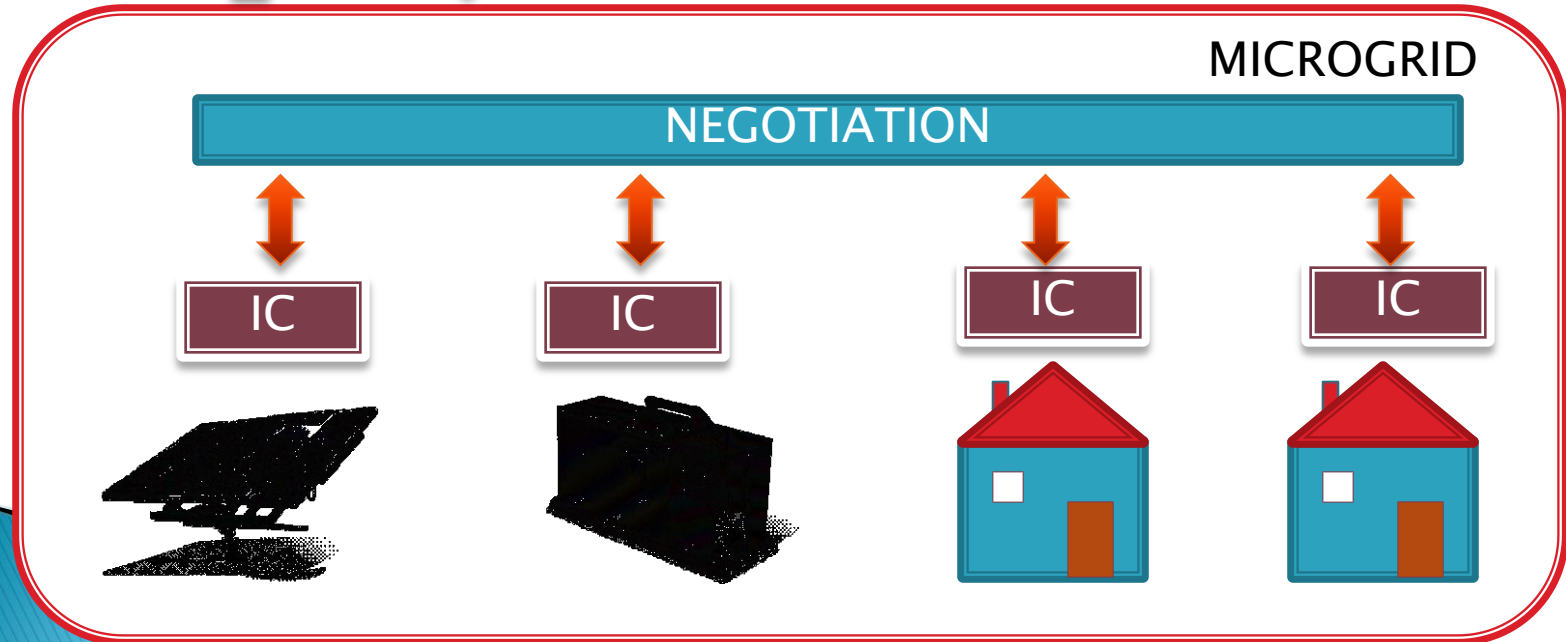


# The Decentralized Approach



Measurements

Price Schedules and Policies



# Implementing the Decentralized Control Concept

- ▶ One approach of implementation adopts the intelligent agent approach
- ▶ Next some basic concepts of the agent theory will be presented as well some practical examples.

# The Agent

Physical entity that acts in the environment or a virtual one

Partial representation of the environment

Agents communicate – cooperate with each other

Agents have a certain level of autonomy

The agents have a behaviour and tends to satisfy objectives using its resources, skills and services

Reactive

partial representation of the environment

autonomy

possesses skills

Cognitive

Memory

Environment Perception

high level communication

# Physical entity that acts in the environment or a virtual one

Physical Entity: Any Hardware that acts into the electrical network!



Virtual Entity: Any (software) entity that interacts with other agents and is part of the system!

DNO

DATABASE

TSO

MARKET OPERATOR

ESCO

WIND FORECAST

ERP

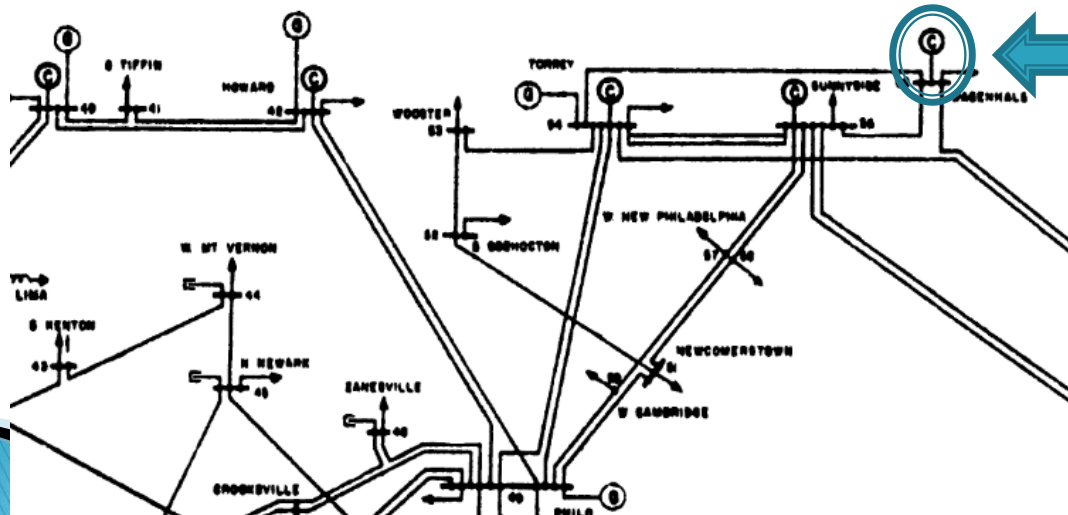
CRM



# Partial representation of the environment

Environment Knowledge: Important for any control system!

- It is very hard to have knowledge of the whole electrical network.
- This is one of the fundamental problems in any power system control application.
- The agent theory suggests that only part of the knowledge may be available in an agent!



The Agent knows electrical values in the connected bus: Voltage, Current, P, Q, frequency....

# Agent Communication

A significant characteristic of agents. The Agent Communication Language allows the interaction and the knowledge sharing.

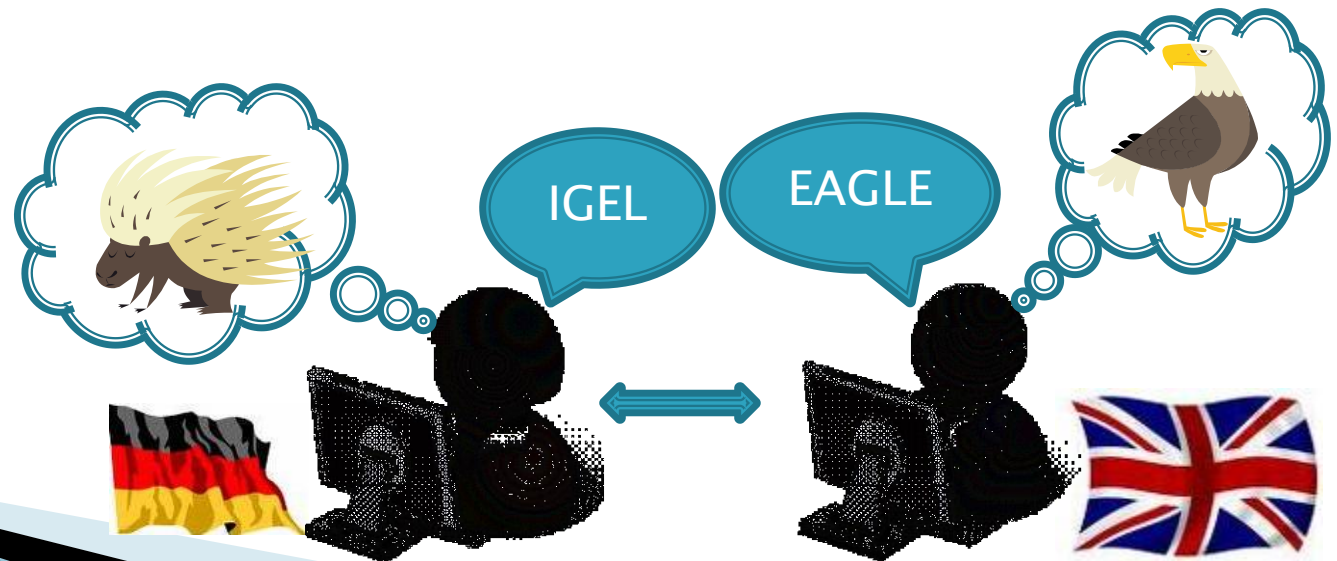
One significant part of the agent communication is the auction algorithm

By adopting the fundamental principals of economic theory, thus fair bidding leads to optimal solutions, the auction algorithm is a useful tool for the agent applications.



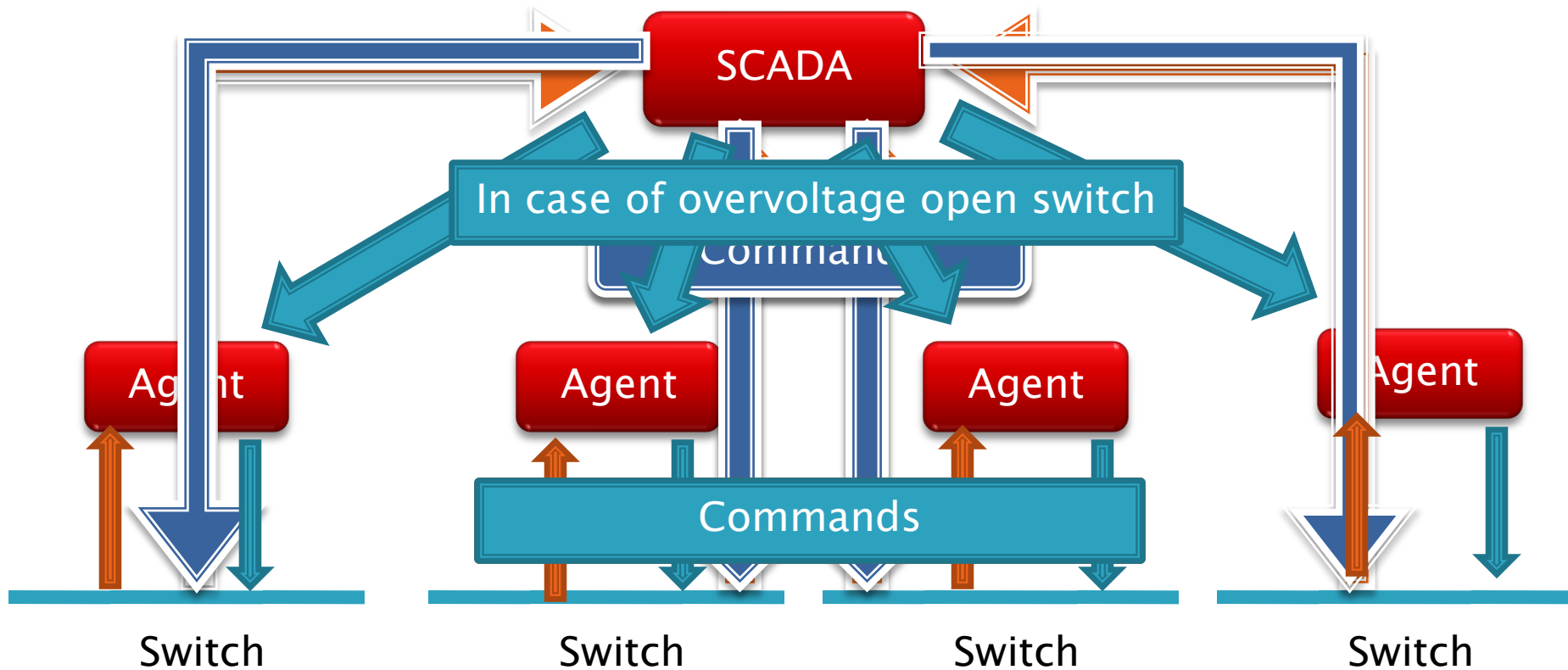
# Agent Communication Language

- The intelligent agent concept requires a strong language capable to describe knowledge
- This language has a structure and a vocabulary called ontology
- The language allows the establishment of complex dialogues



# Agents have a certain level of autonomy

This characteristic is a consequence of the local intelligence!



# Behaviour ,objectives, resources, skills and services

## Behavior

- Competitive
- Collaboration

## Objectives

- Maximize profit
- Minimize cost

## Resources

- Available Fuel
- Energy Stored in a Battery

## Skills

- Load Curtailment
- Black Start

## Services

- Yellow pages
- Data Storage

# Reactive vs Cognitive

## Reactive

- ▶ The agent react to certain signals
- ▶ The collaboration of several reactive agents may form a intelligent society
- ▶ Typical example: the ant colony
- ▶ For an electrical network a protection device is a reactive agent.
- ▶ Several protection devices may create a self healing network



## Cognitive

- ▶ The agent has increased intelligence and advance communication capabilities.
- ▶ The collaboration is supported by the intelligence and the communication capabilities
- ▶ Typical example: the human society

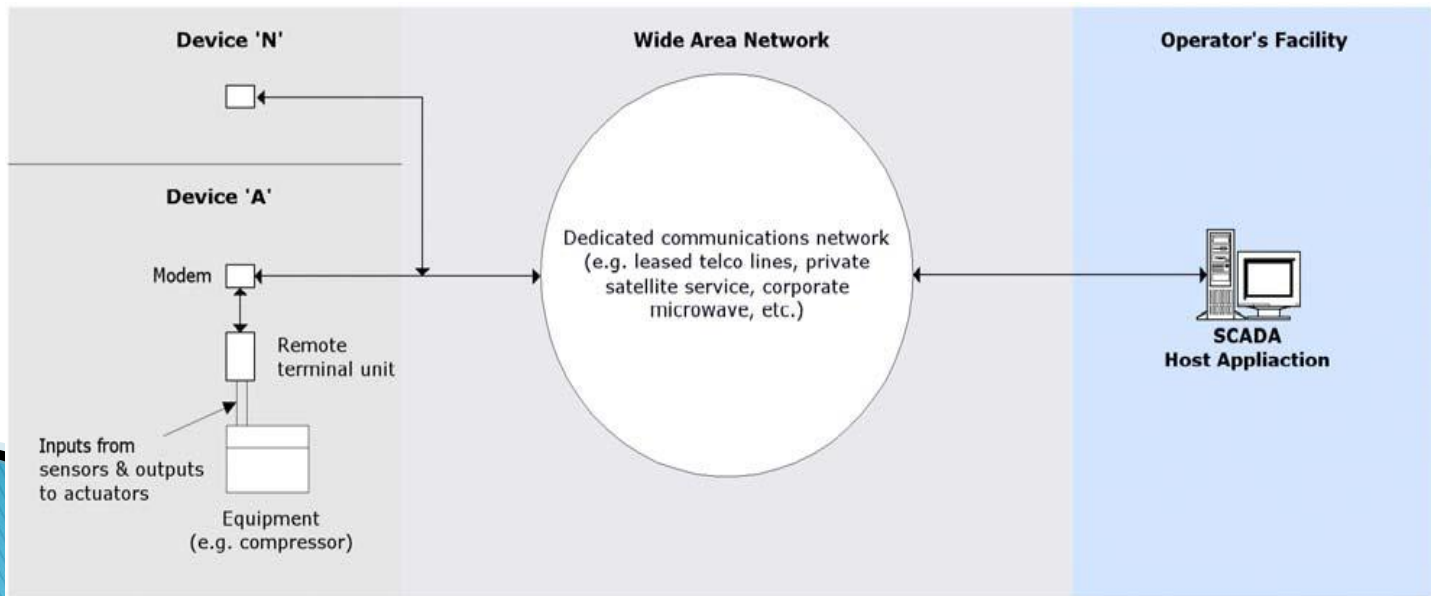


# The Centralized Control

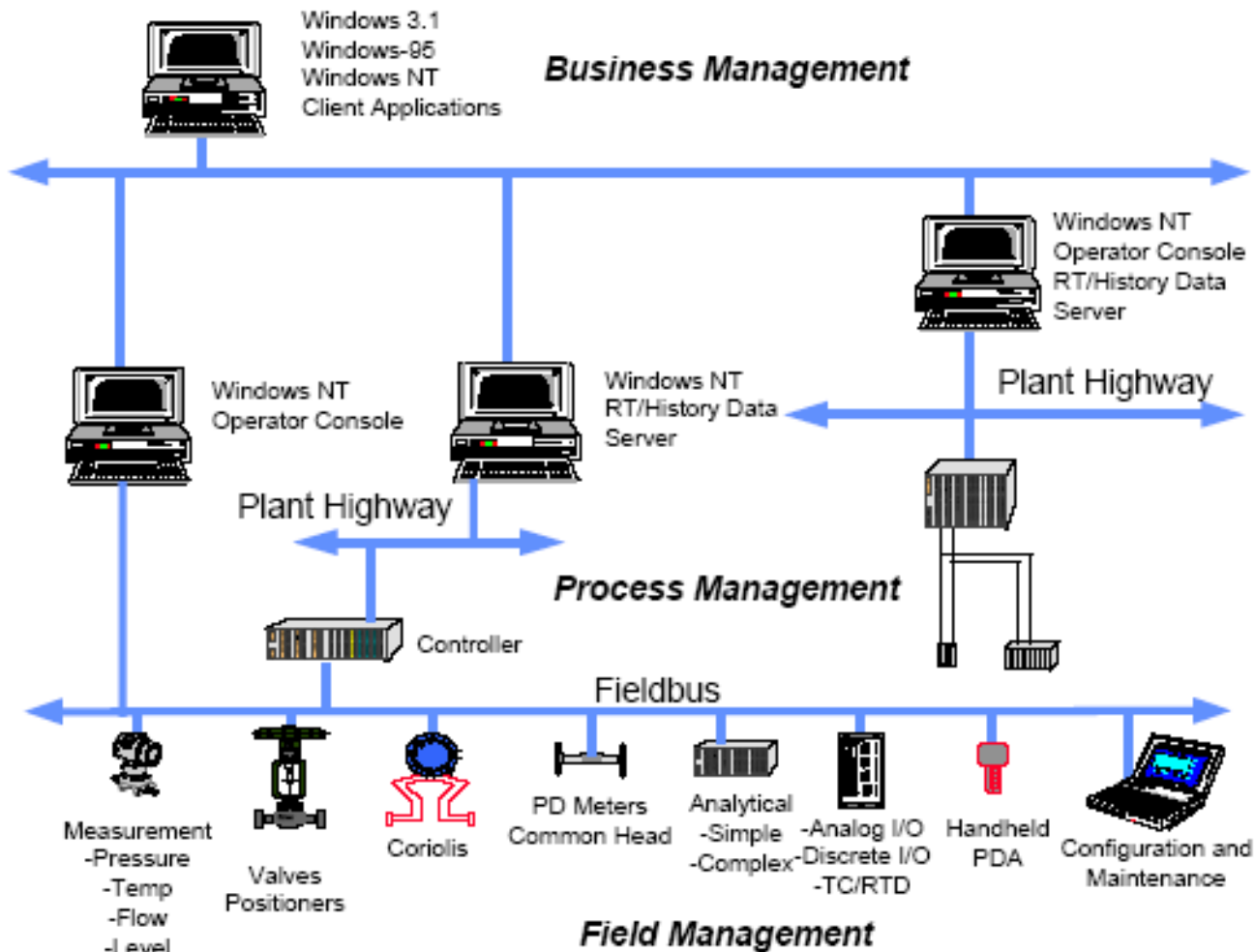


# SCADA

- ▶ The Supervisory Control and Data Acquisition (SCADA) is a set of software for controlling and monitoring equipment and processes
- ▶ Can be used to support the Microgrids Operation

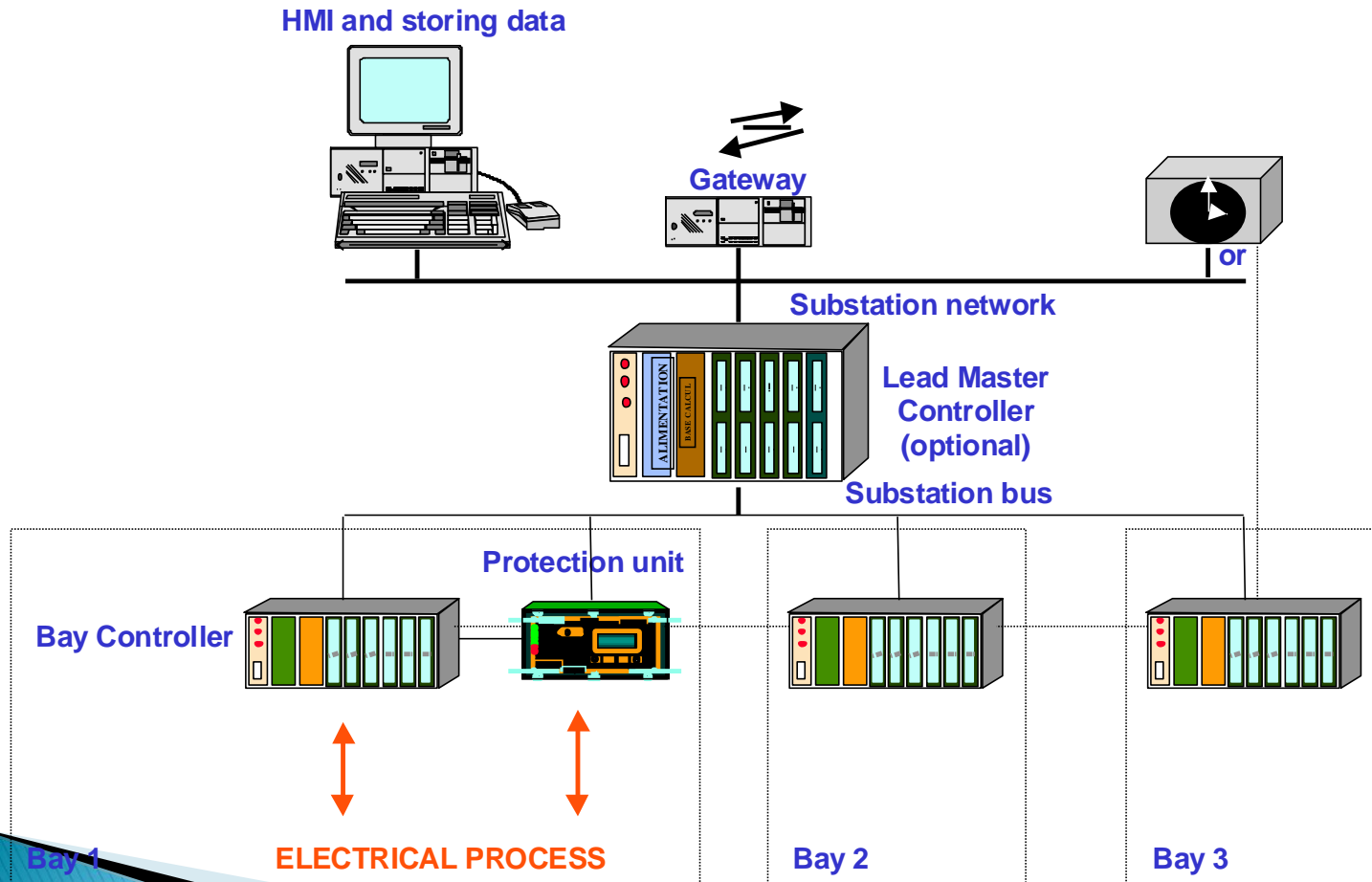


# Structure of Typical Control Applications



# Substation Control Structure

## EUROPE

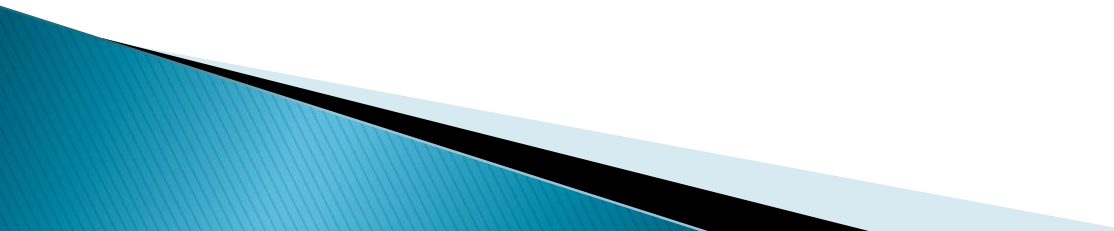


# Examples of Implementations

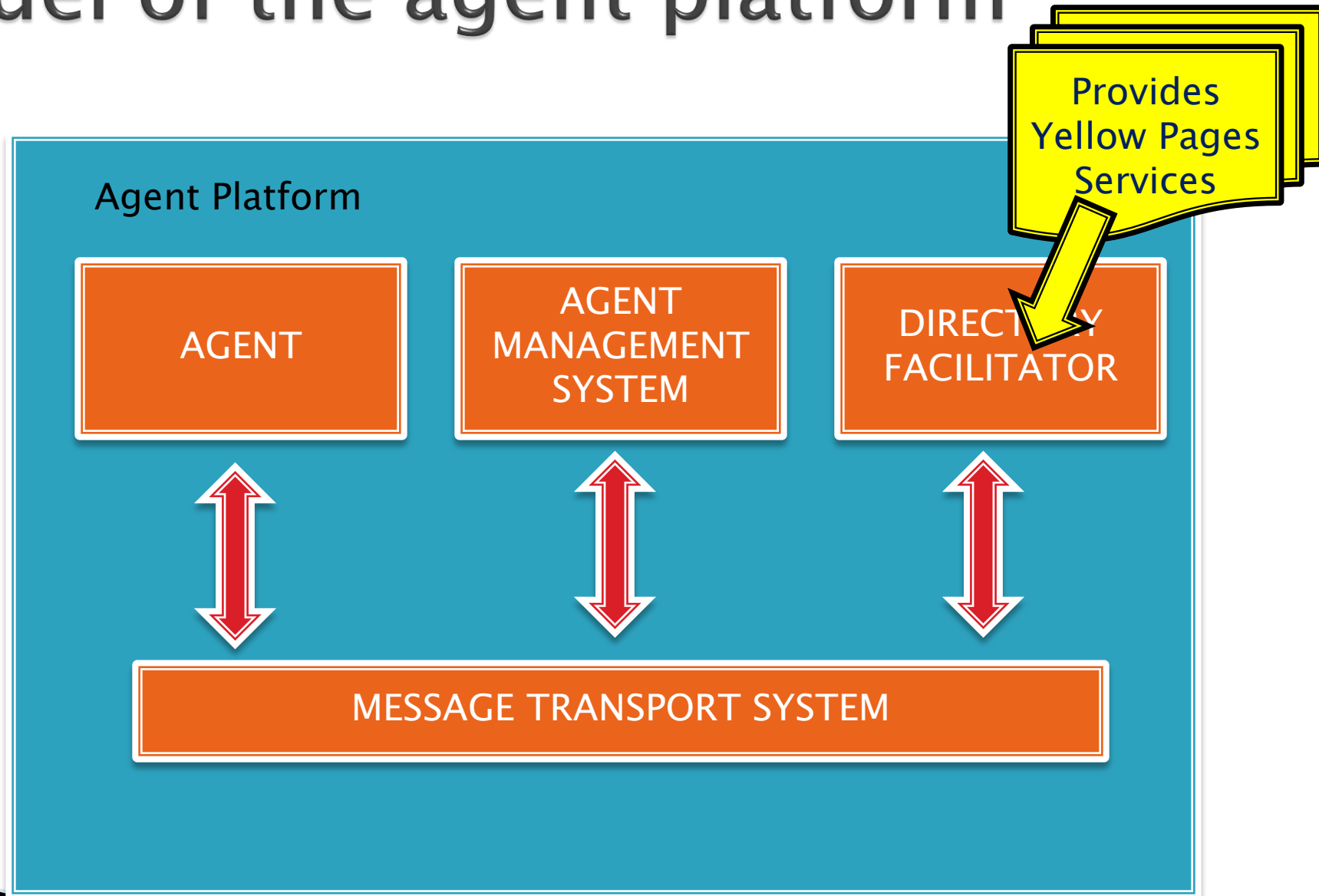


# Example #1

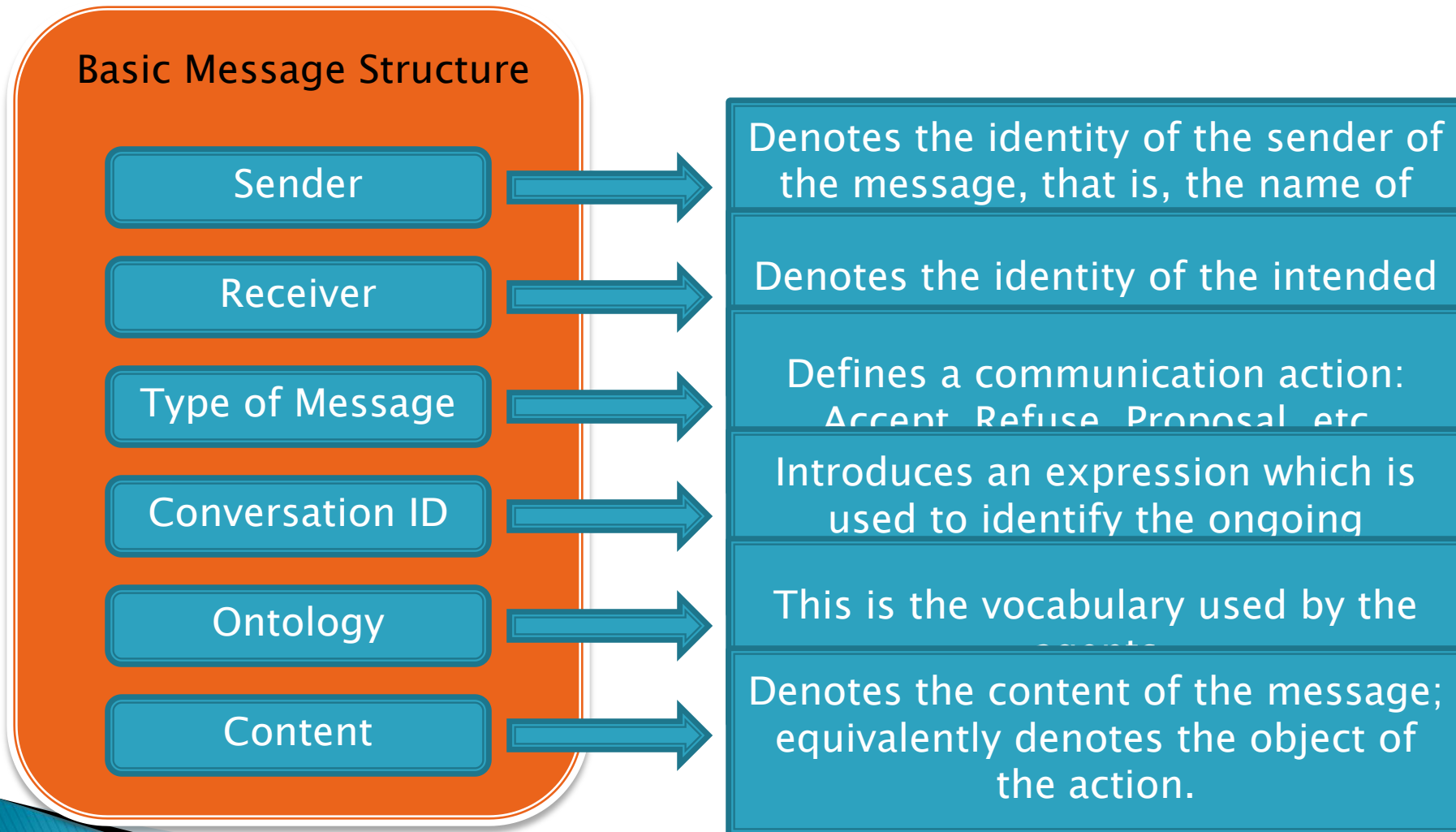
## Implementing Agent with Java-Jade

- ▶ Jade is a Java based platform for agent implementation.
  - ▶ It is compatible with FIPA requirements
  - ▶ FIPA is the Foundation for Intelligent Physical Agents
  - ▶ Jade provides a set of libraries that allow the implementation of the agents.
- 

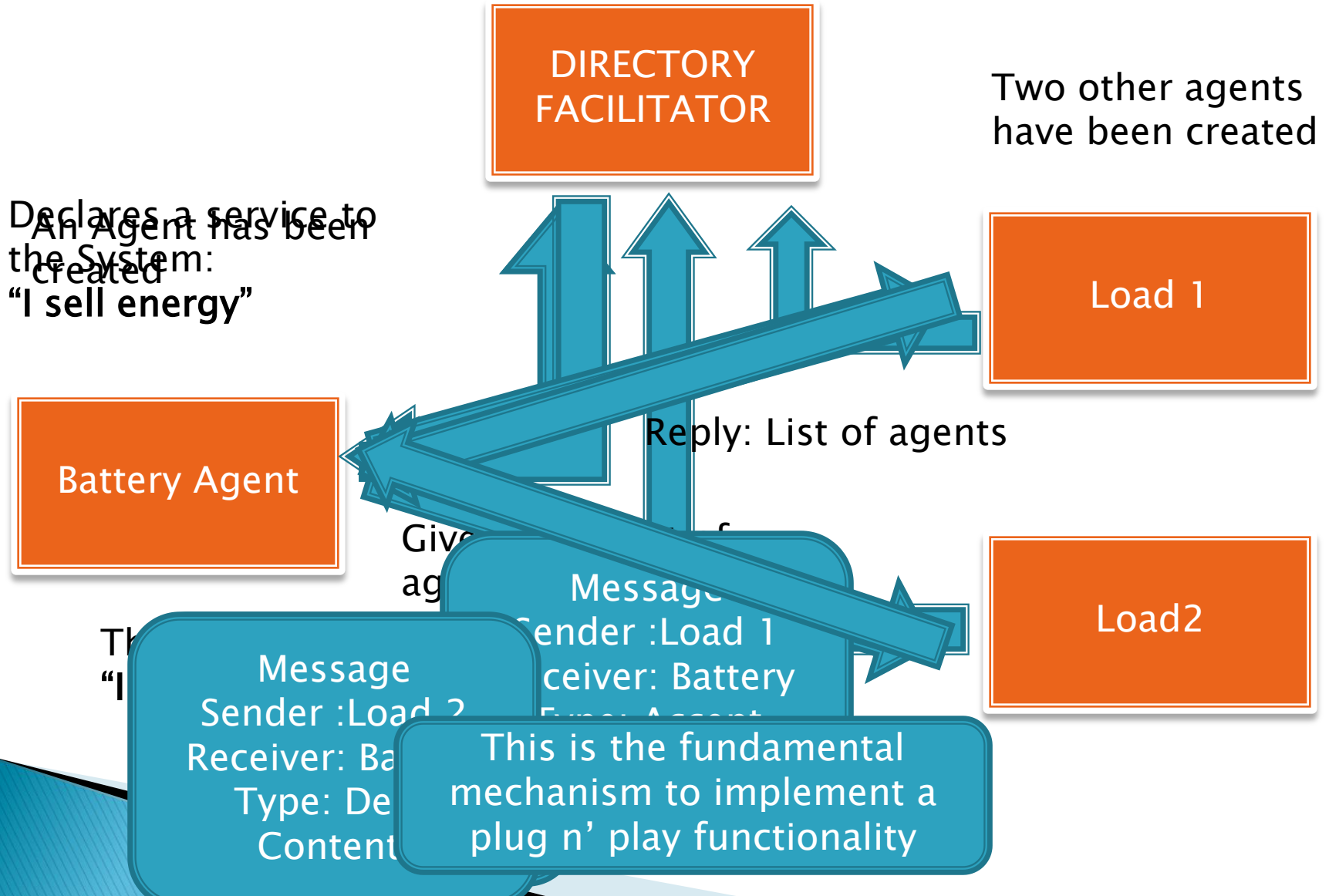
# Model of the agent platform



# Implementation of the dialogues

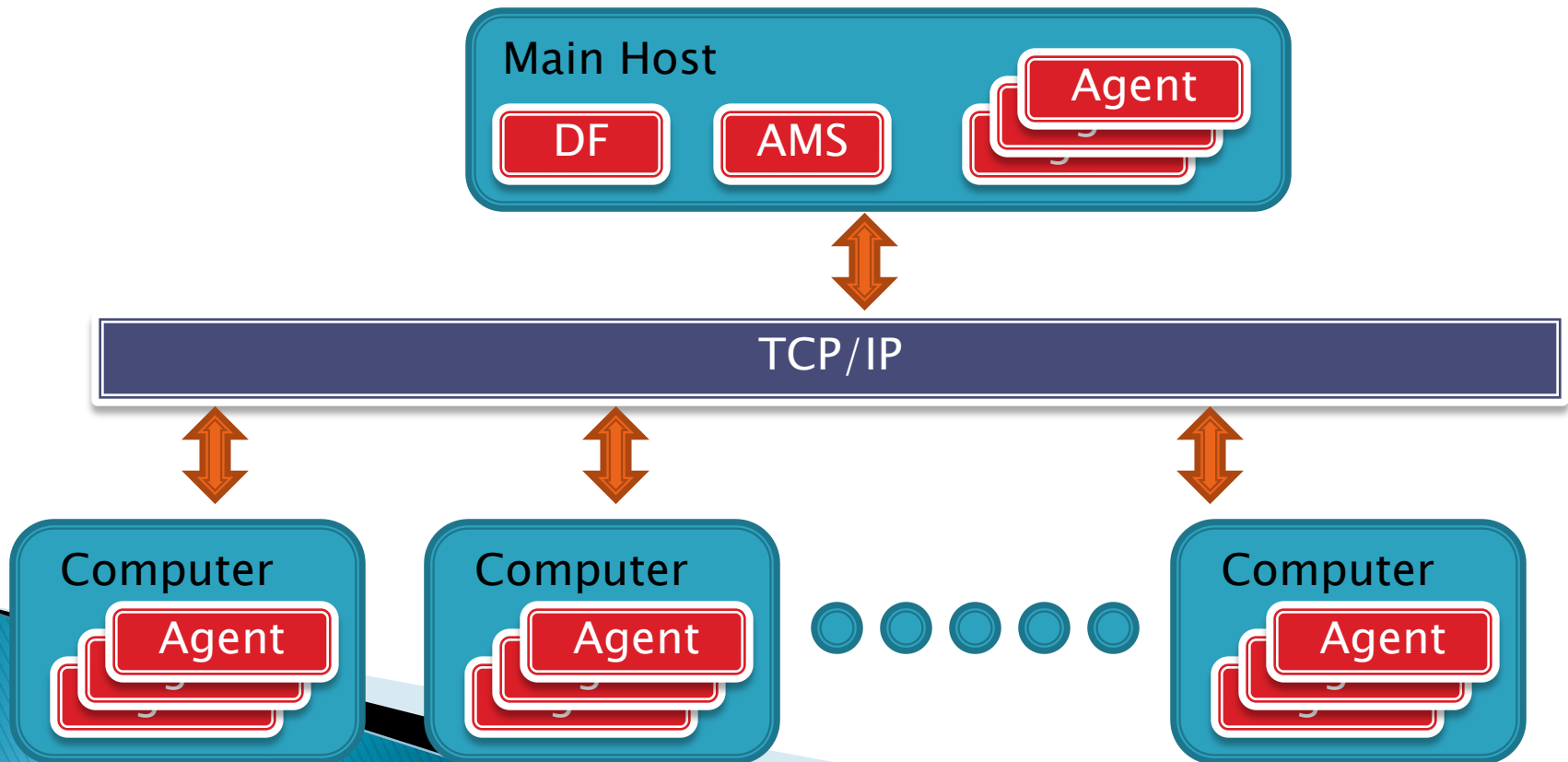


# Simplified Conversation



# Implementation

- ▶ The Implementation of the system requires a network of computers capable to host Java Applications and communicate over TCP/IP (internet protocols)



# Example #2

- ▶ Installation of the agents and the Siemens V6.0 WinCC SCADA in the Laboratory

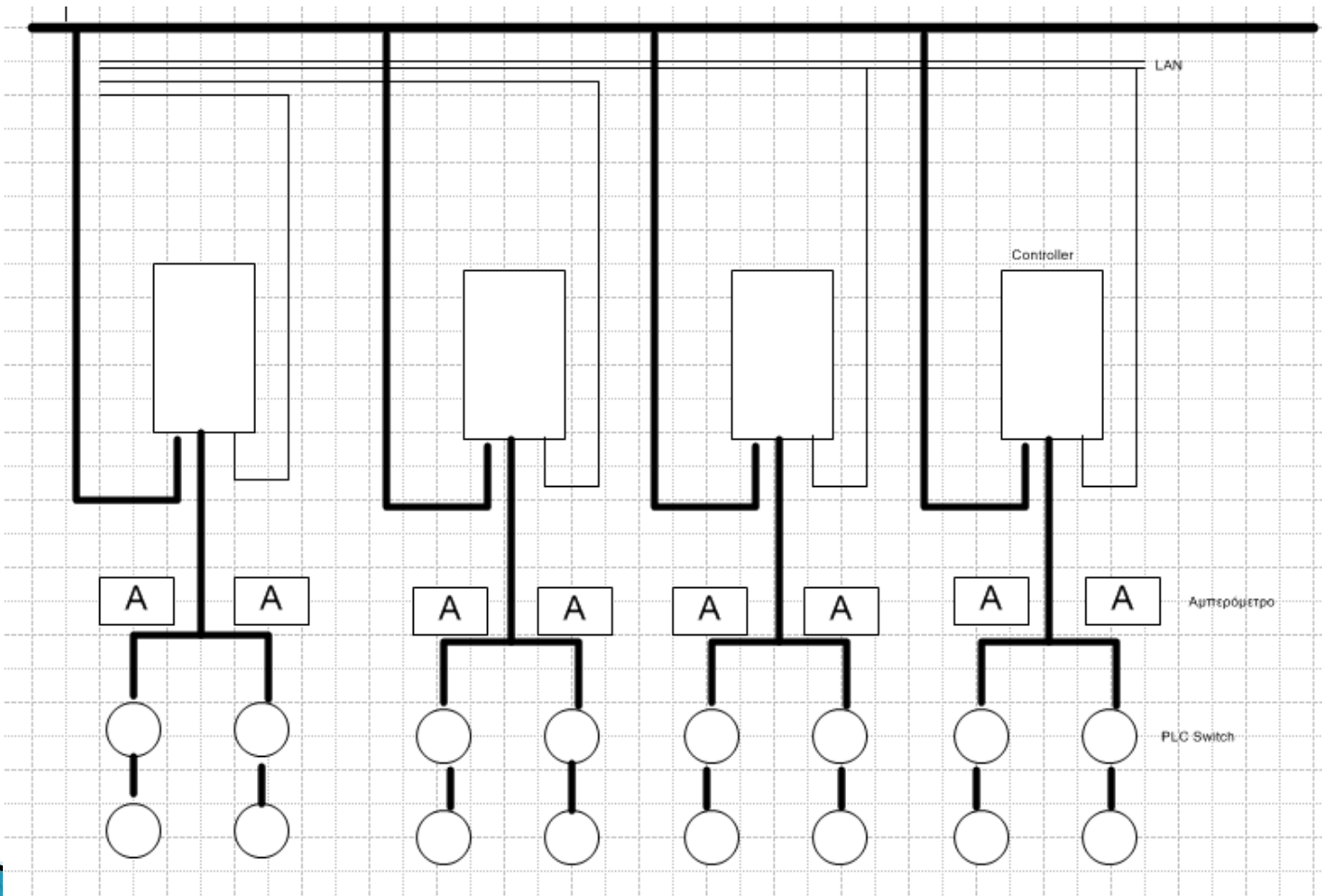
# Laboratory Overview



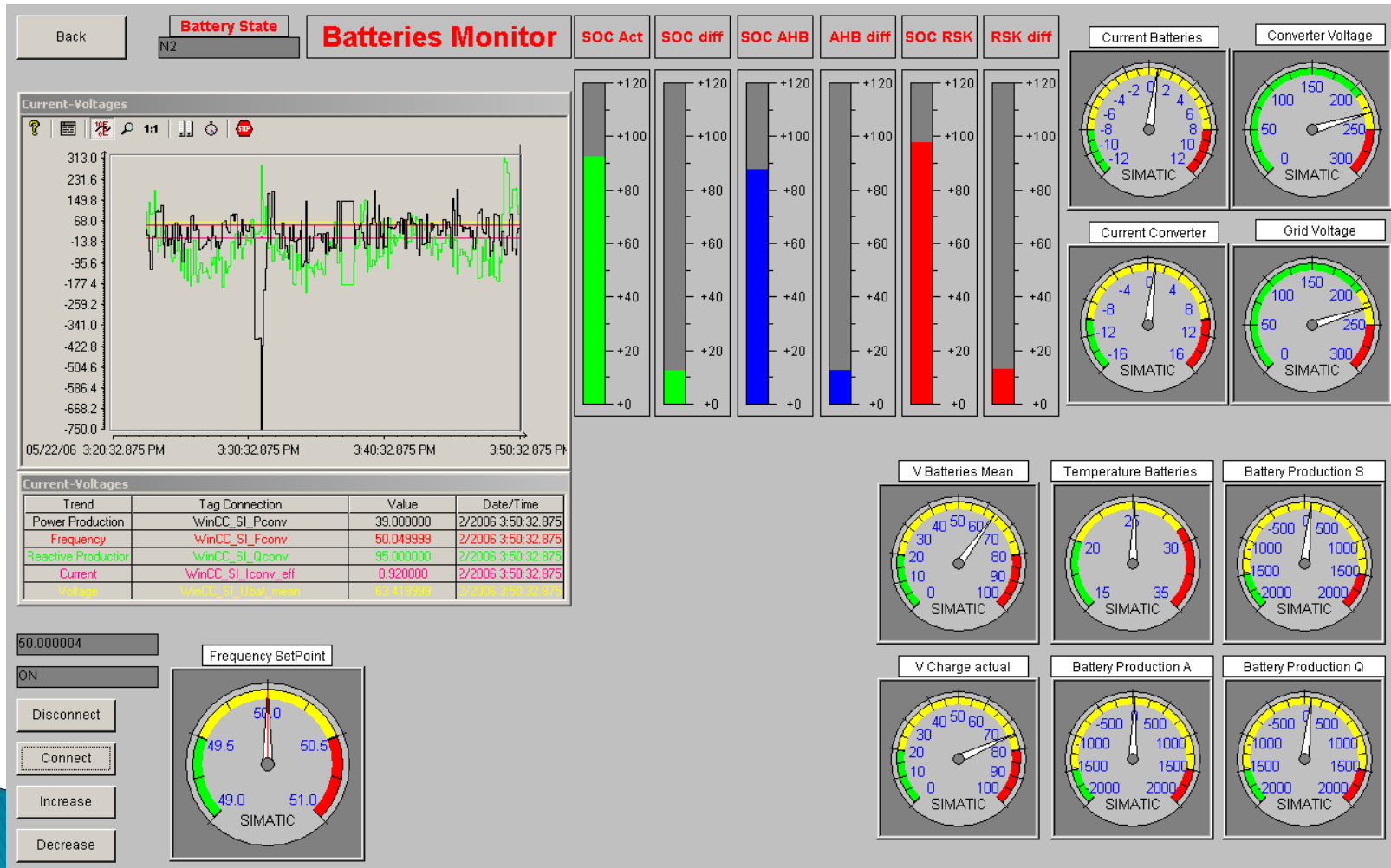
# Panel with Load Controllers



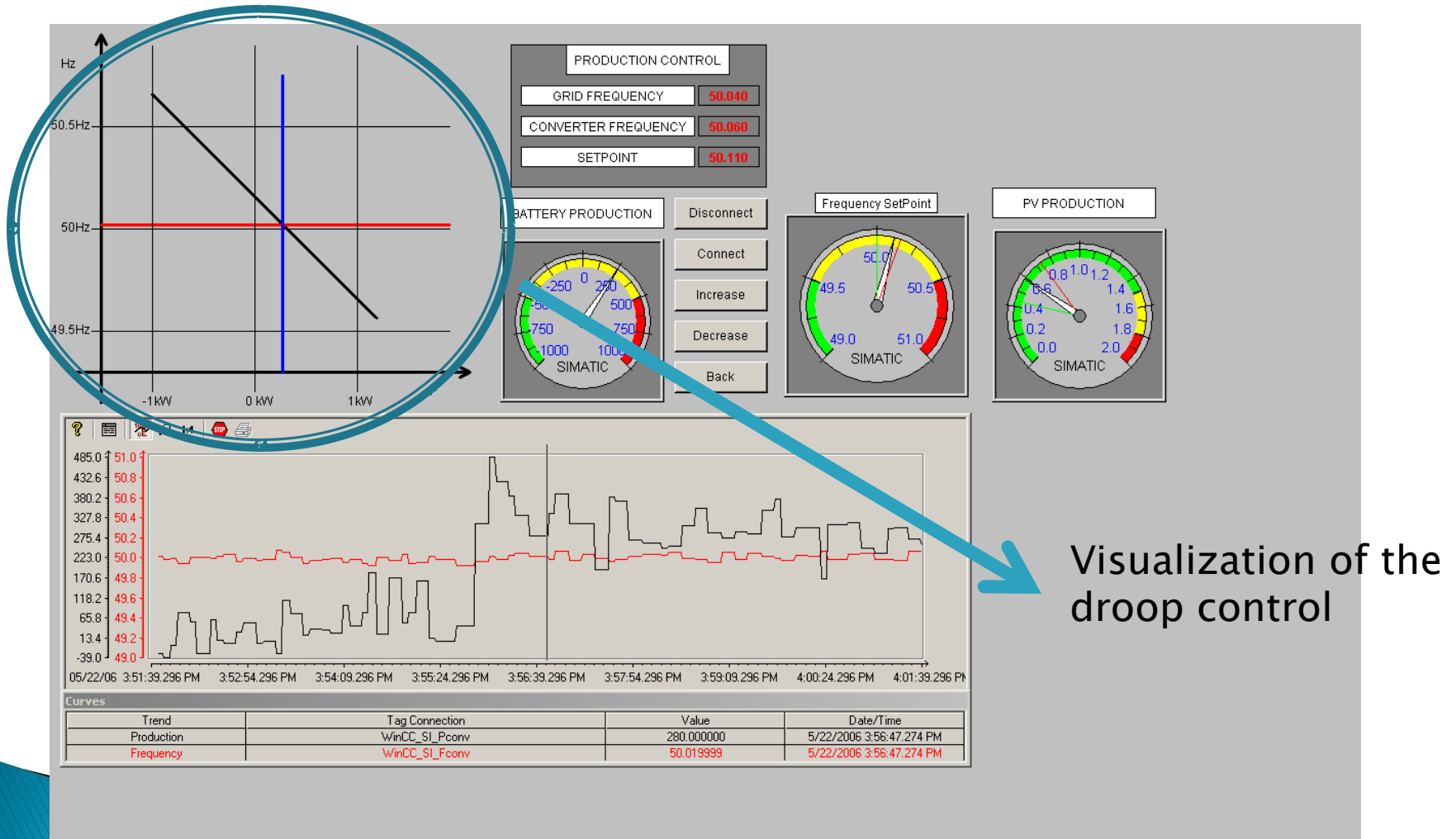
# Schematic of the Panel



# The MMI for the Microgrid Laboratory



# The MMI for the Microgrid Laboratory



Visualization of the droop control

# Screenshot of the Agent implementation

The image displays two software windows. The 'MG Central Controller' window is on the left, and the 'PV Panel' window is on the right.

**MG Central Controller**

- Status:** Market, StandBy, BlackStart, ShutDown (highlighted in red).
- Power Control:** -600
- SI production:** 0.0
- SB production:** 0.0
- Controls:** Voltage Control, Frequency Control, Reactive Control, Power Control, Black Start (all unchecked).
- Monitor:** MGCC 2 Apr 2006 19:20:51 GMT  
Voltage = 0.0 V  
Frequency = 0.0 Hz  
Grid Connected = 0  
SI Status = 0  
SI Mode = 0  
SI Production = 0.0 W  
SI Reactive Production = 0.0 VA  
SB Production = 0.0 W  
SI Battery SOC = 0.0 %
- Messages:**  
Offer for production : from Battery price 12.0  
Offer for production : from PV Unit price 14.0  
Offer for production : from Main Grid price 10.0  
Found New Load : ( agent-identifier :name House#2@aris-laptop:1099UADE :  
Found New Load : ( agent-identifier :name House#1@aris-laptop:1099UADE :  
Init Parameters send toHouse#2  
Init Parameters send toHouse#1  
Main CREATED NEW AGENT: PowerSystem from microgridload.PowerSystem  
Main CREATED NEW AGENT: TrackingUnit from microgridload.TrackingUnit  
Main CREATED NEW AGENT: House#2 from microgridload.LoadsUnit

**PV Panel**

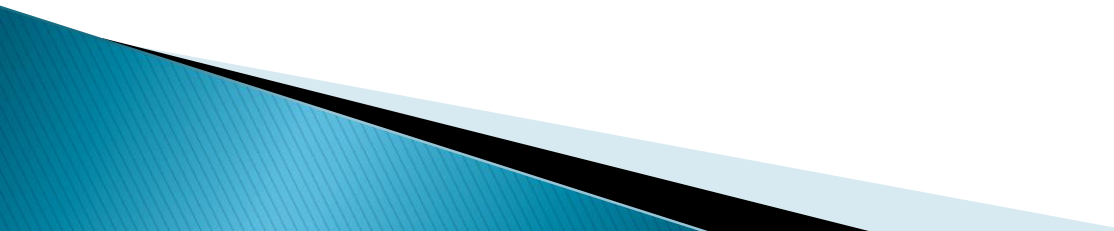
- Unit Production (W):** 0
- Unit Max Production (W):** 0.0
- Operational Cost (EuroCent/kWh):** 14
- Set Point (W):** 0
- Set** button
- Messages:**  
C:>  
PV Unit Launched  
Market Started

## Example #3

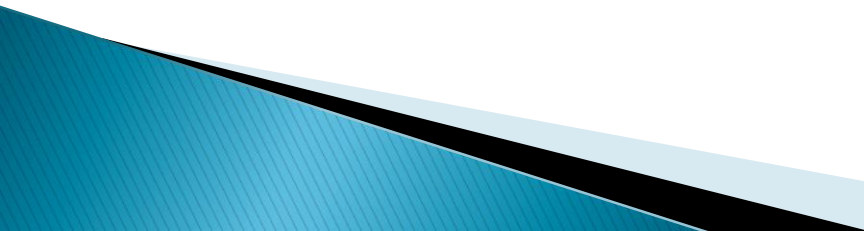
# Agent Based Control in a Kythnos

- ▶ This is our first implementation of the Agent Based Control
- ▶ The installation is in the island of Kythnos
- ▶ The goal is to optimize the consumption of the houses

# More Microgrids Objectives

- Sophisticated control techniques for Distributed Generators Load Controllers to implement
  - Study of integration of several Microgrids into operation and development of the power system. Interaction with DMS.
  - Need for standardization and benchmarking.
  - **Field trials to test control strategies on actual Microgrids.**
  - **Need for quantification of Microgrids effects on Power system operation and planning.**
- 

# Goal of the Experiment

- ▶ The Goal of the experiment is to test the agent based control system in a real test site in order to increase energy efficiency.
  - ▶ The main objective is to test the technical challenges of the MAS implementation.
  - ▶ The technical implementation will be based on the load controller described in a previous Presentation and the Jade Platform
  - ▶ The algorithm regarding the increase of the energy efficiency is quite simple and focuses in the limitation of the pump operation.
- 

# Goals of the Experiment

## Software

- Java/Jade implementation
- CIM based ontology

## Hardware

- Embedded Controller
- Measurements
- Communication
- Control via PLC

## Technical

- Implement Distributed Control
- Test in real Environment

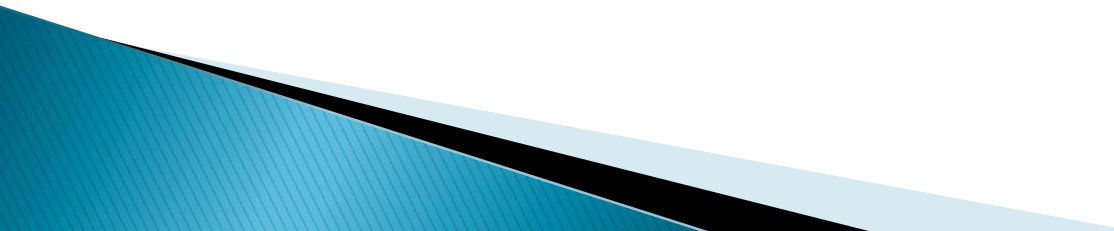
## Electrical

- Increase energy efficiency
- Manage Non Critical Loads

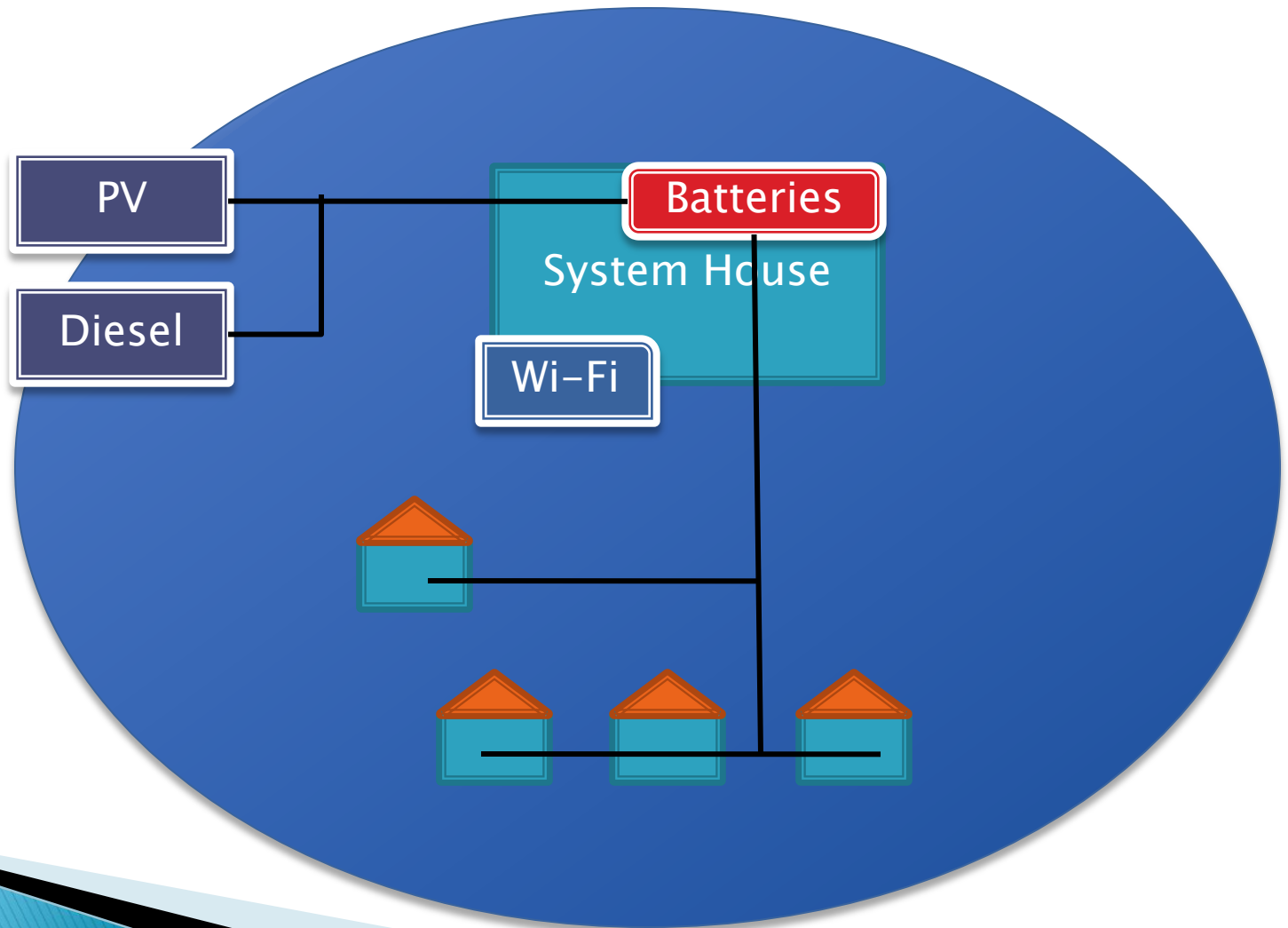
# Short Description of Kythnos Installation

- ▶ The test site is a small settlement with 13 houses

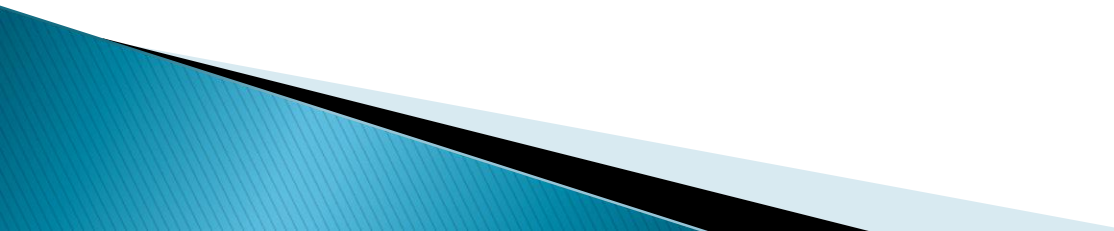
The production system includes:

- ▶ 12 kW of PV
  - ▶ 53-kWh batteries
  - ▶ 5-kW diesel
- 

# General Configuration

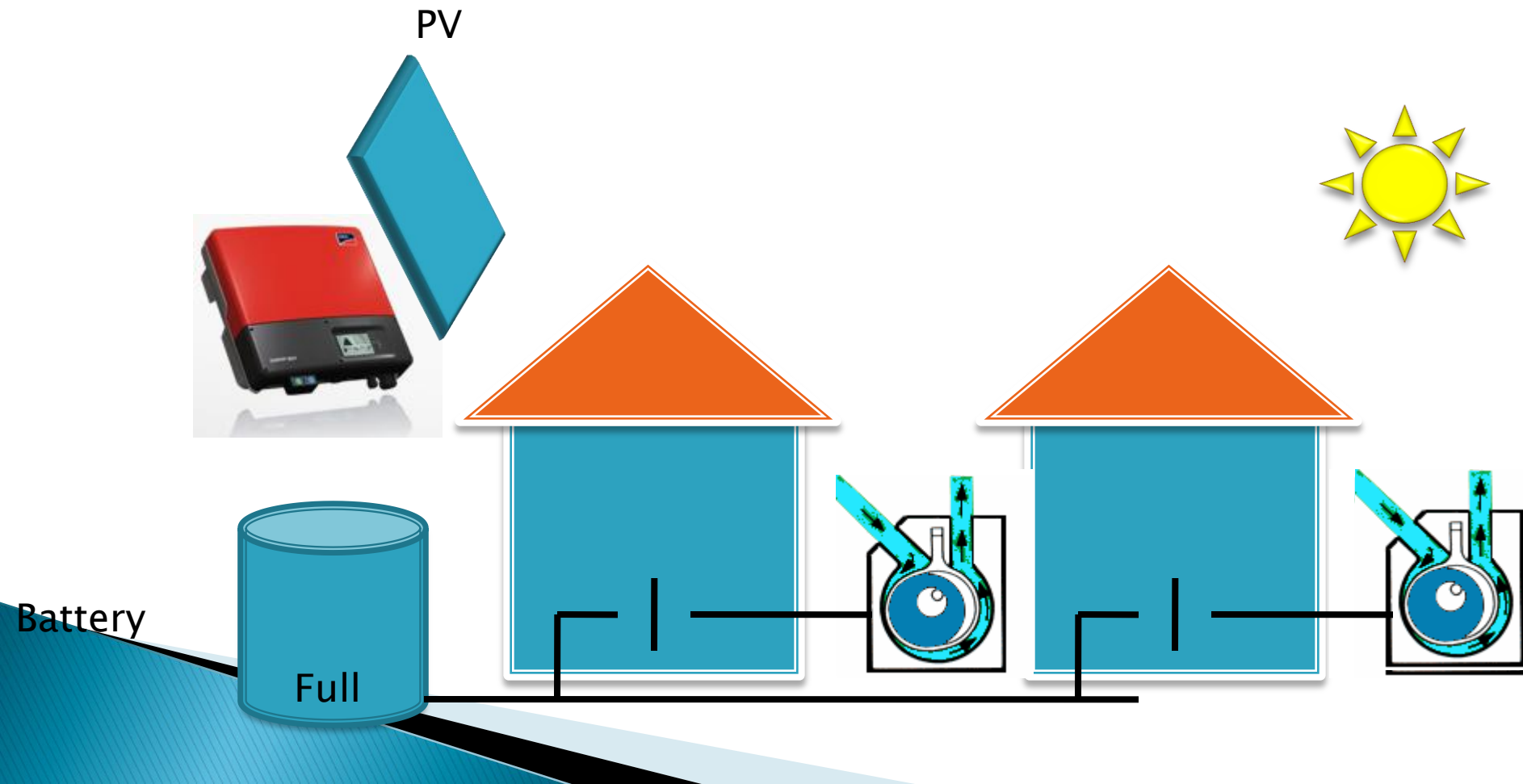


# The MAS System

- ▶ The MAS tries to increase the energy efficiency. The steps are the following:
    1. The system decides the available energy that can be used by the pumps.
    2. The houses decide how to share this energy.
- 

# The general idea:

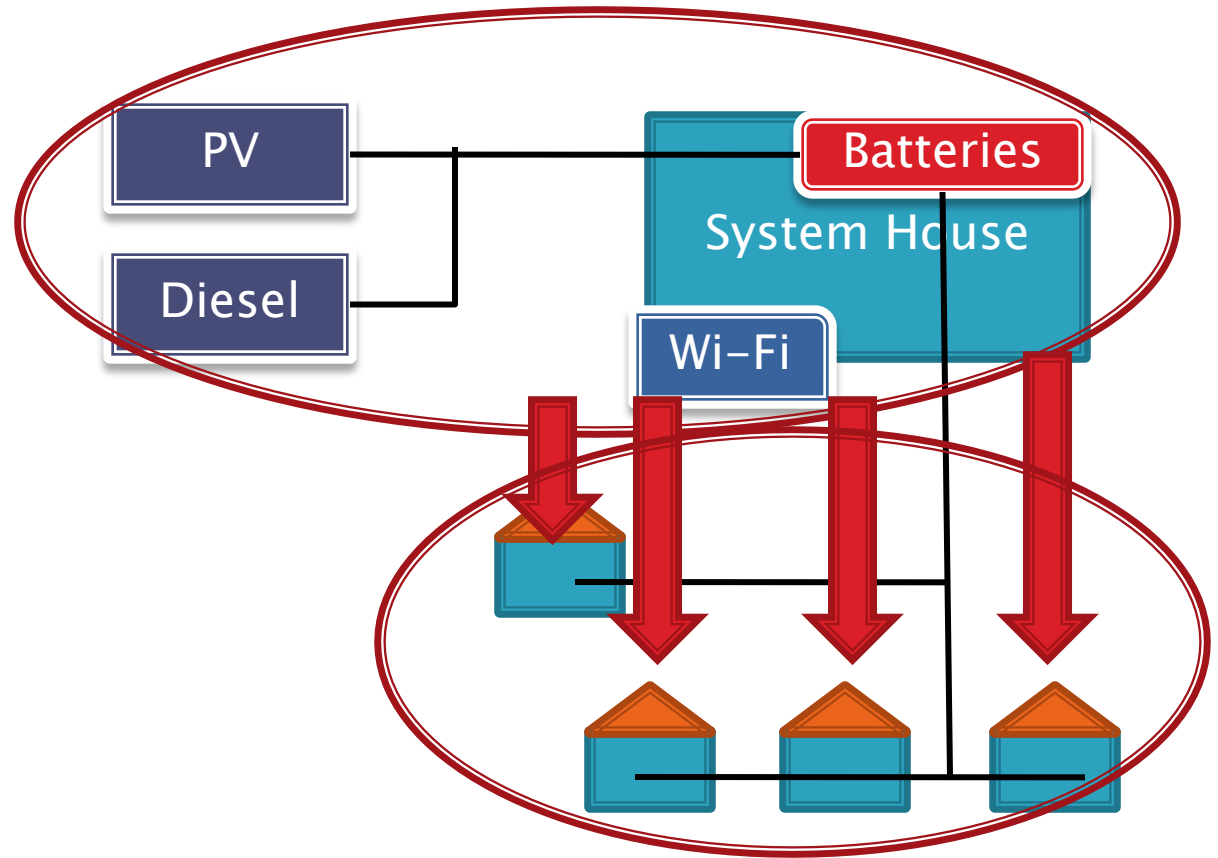
The main load in each is the water pump.  
The goal of the system is to limit the usage of the pumps



# The Process of the experiment

Step 1: The agents identify the status of the environment

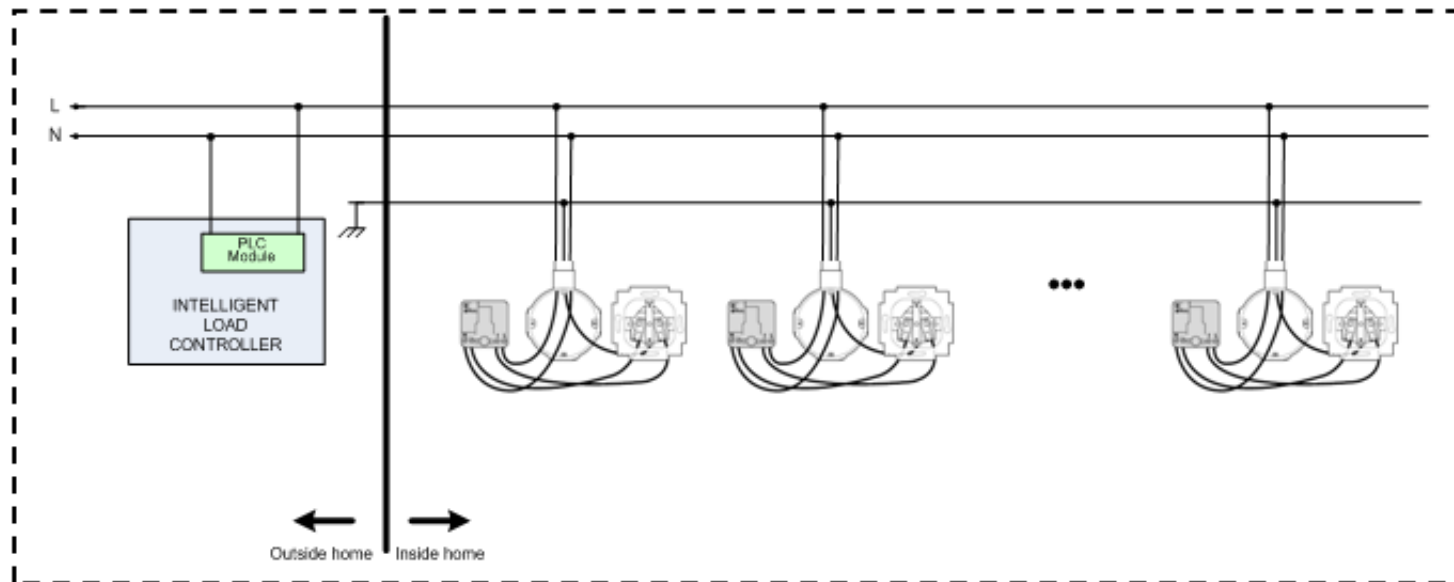
Step 2: The agents negotiate on how the share the available energy

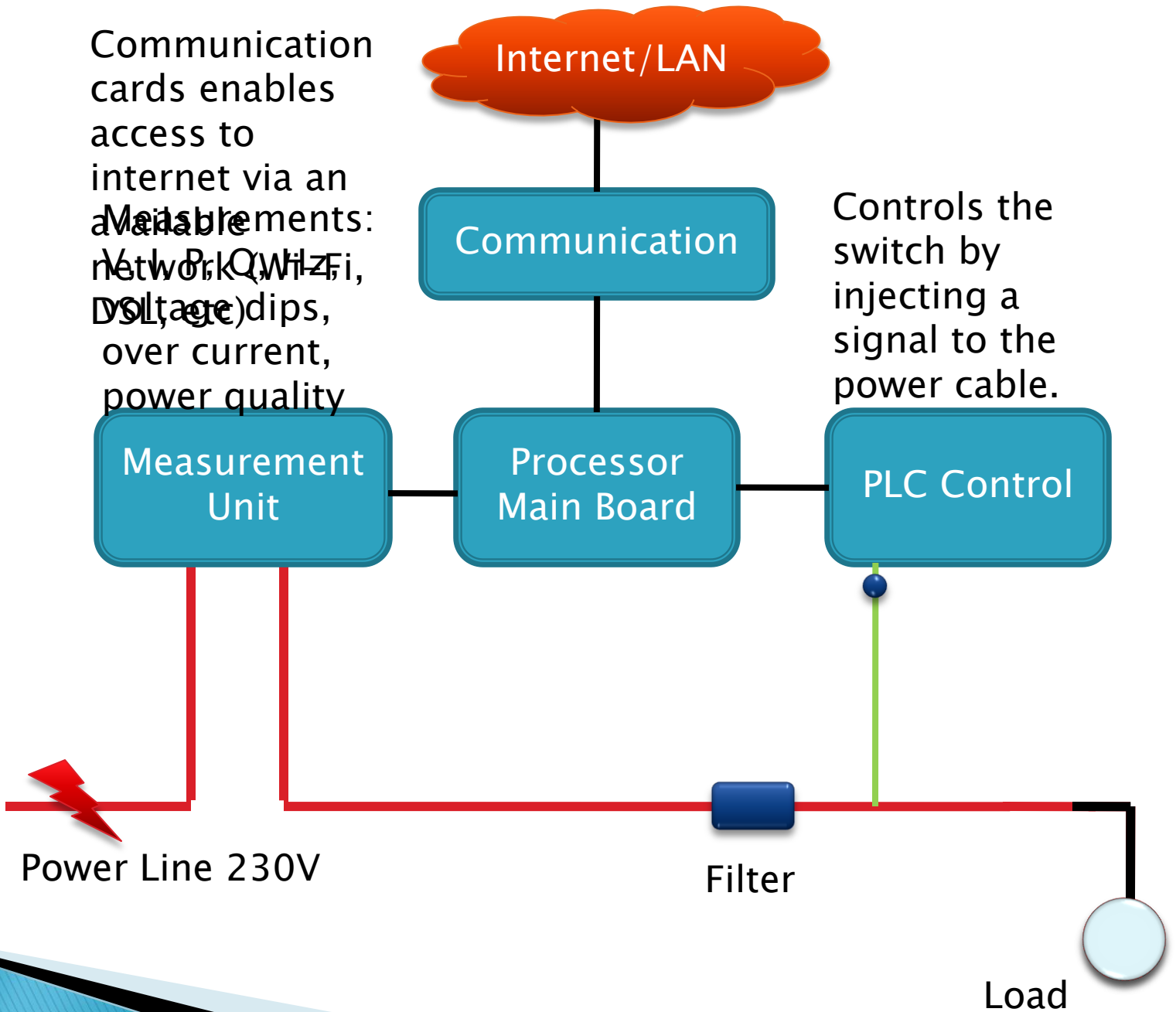


# The Load Controller: a critical component

In each house an Intelligent Load Controller has been installed:

1. Windows CE 5.0
2. Intel<sup>®</sup> Xscale<sup>™</sup> PXA255
3. 64MB of RAM
4. 32MB FLASH Memory
5. Java VM
6. Jade LEAP





# Installation of the Controllers

Outside System House



House 11



House 7



Inside System House



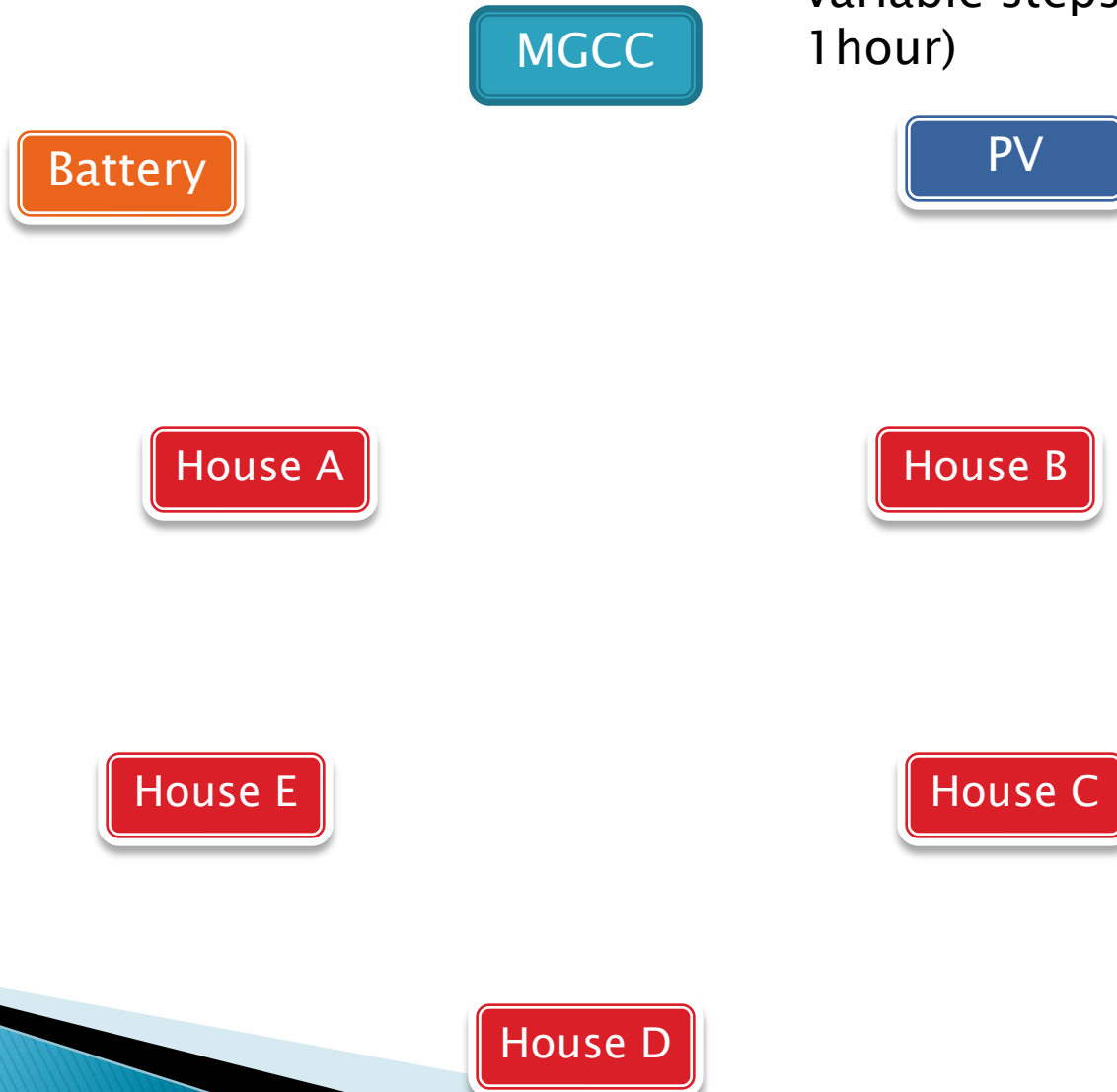
House 5



House 4



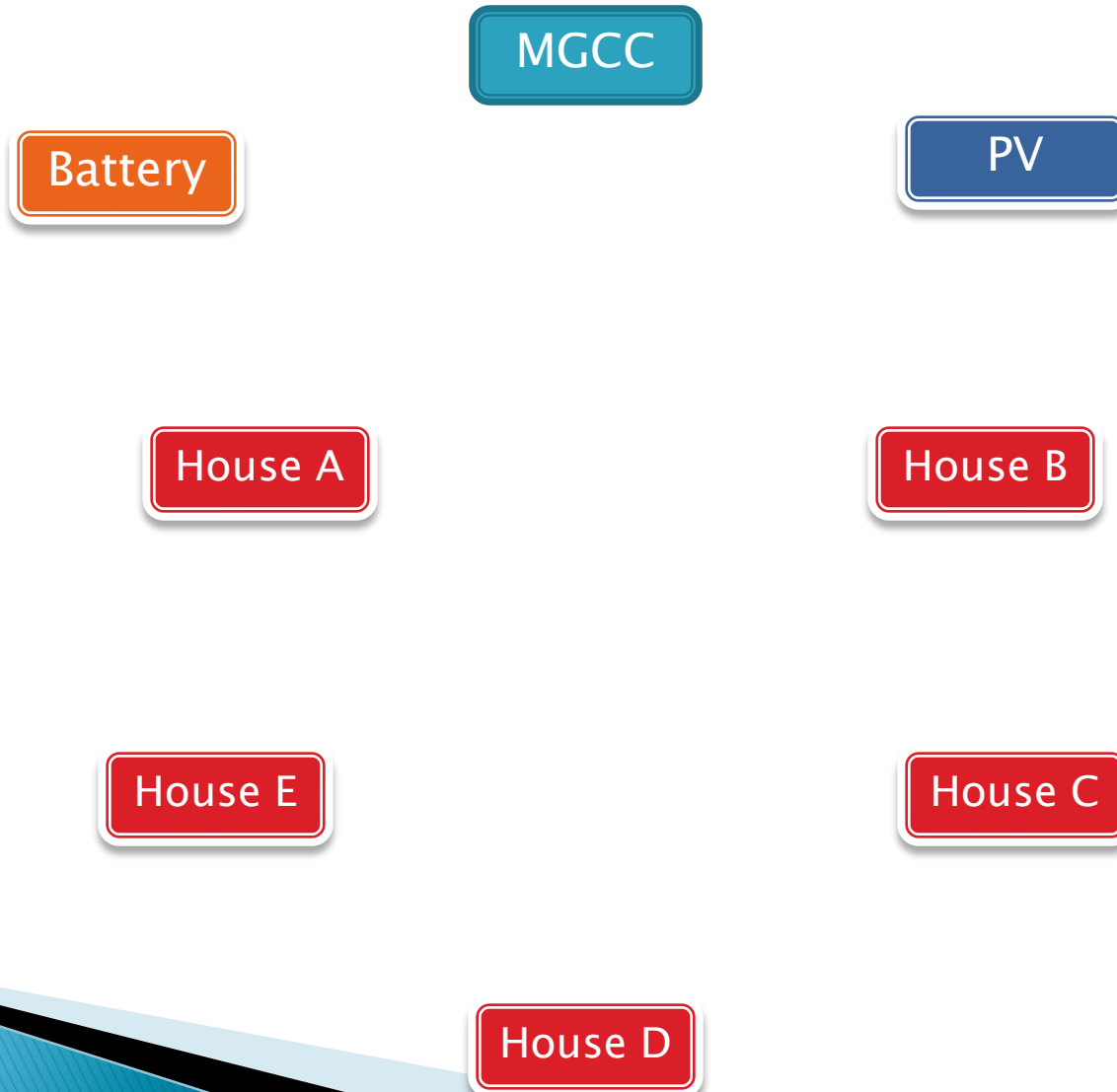
# Negotiation



Start Negotiation.  
The MGCC orders the system to start a new cycle. This can be done in variable steps (5min-1 hour)

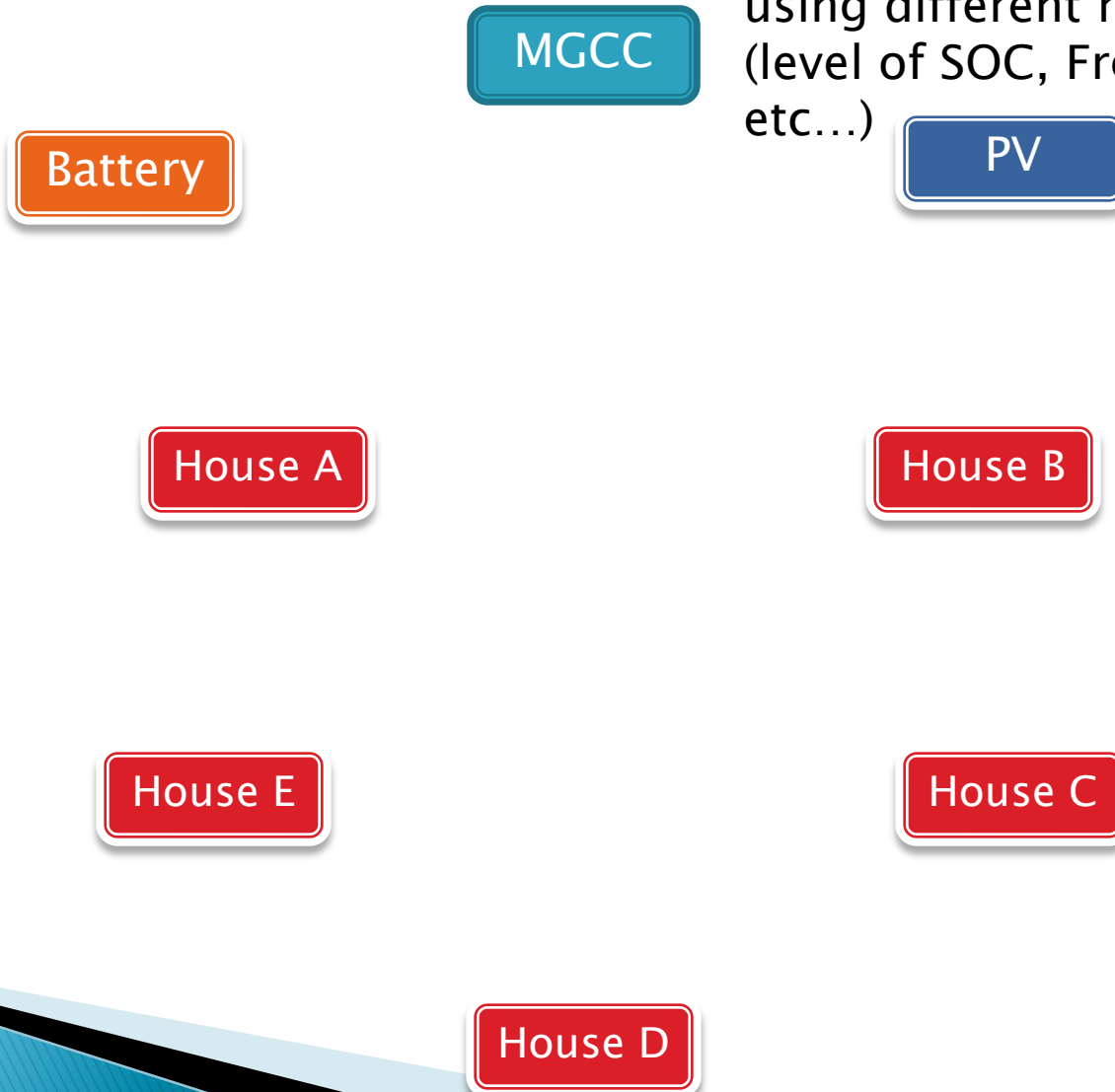
# Negotiation

PV agent Announces  
Production



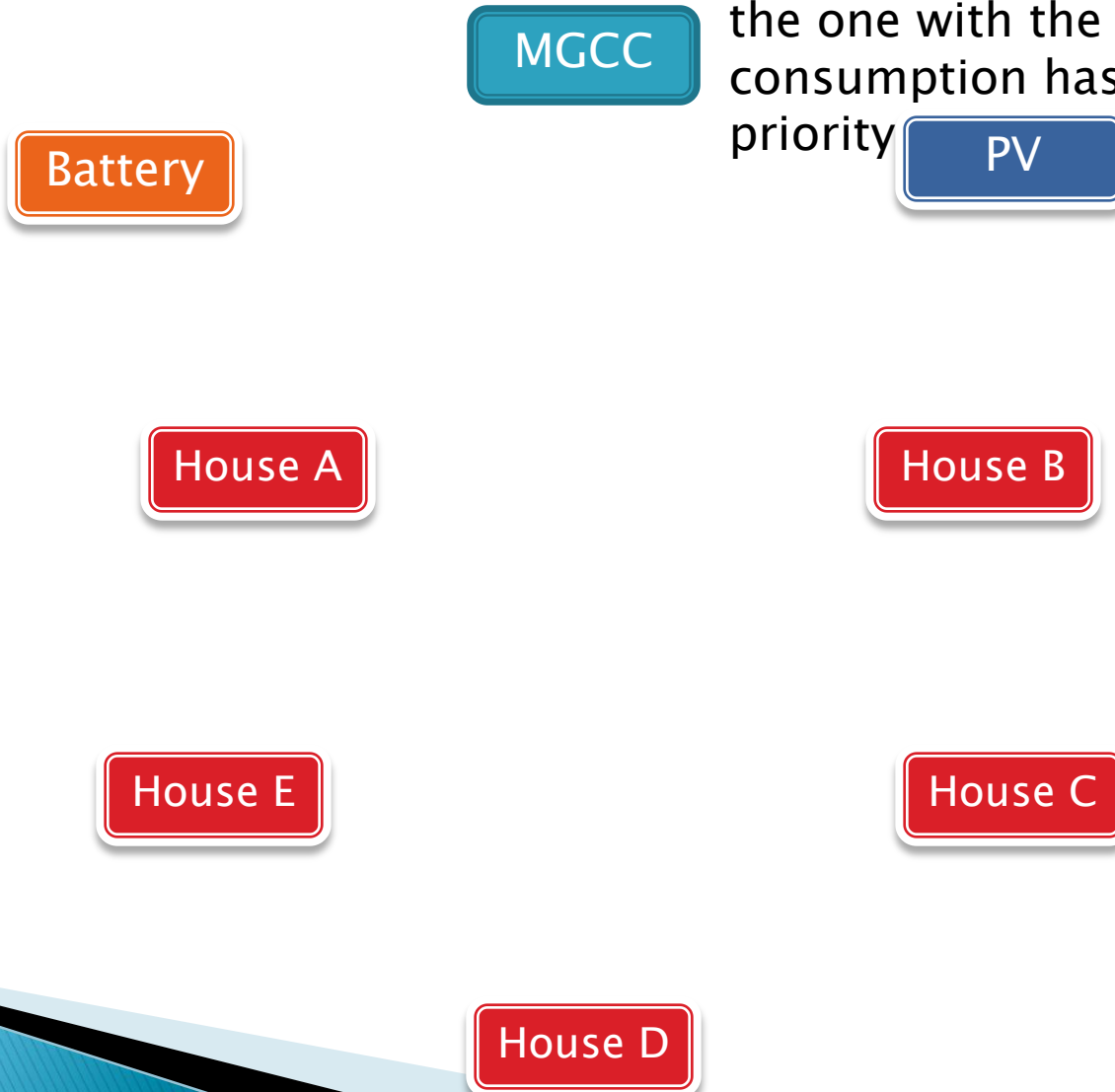
# Negotiation

Battery agent Announces Production & SOC.  
The estimation of the available energy can be done using different methods (level of SOC, Frequency, etc...)



# Negotiation

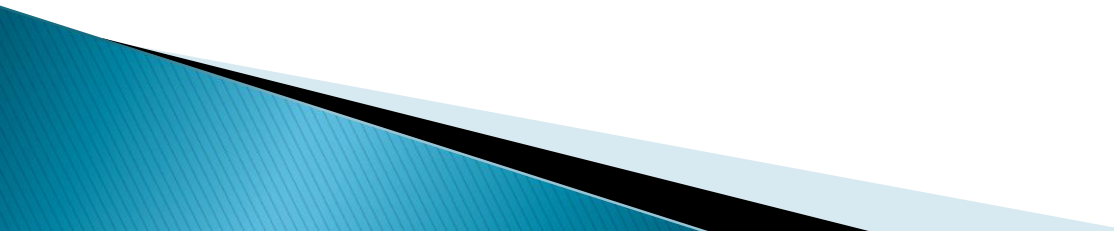
Agents Start Negotiating.  
The simple algorithm suggests that agents should consume equally. Therefore the one with the higher consumption has the lower priority



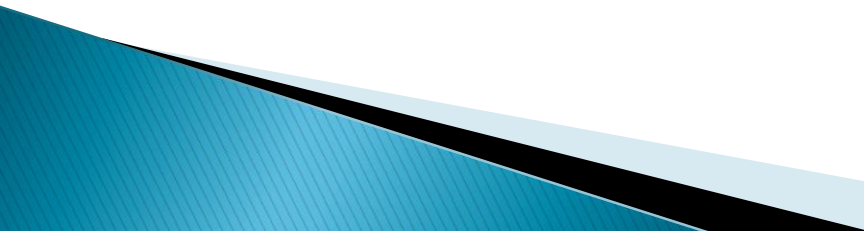
# Theory

- ▶ Auction algorithms

# Auction Algorithm

- ▶ One significant part of the agent communication and decision process is the auction algorithms.
  - ▶ The auction algorithm is a tool that allows the agents to decide which one of them will acquire a certain object or a good.
  - ▶ Next an algorithm will be presented : the English Auction
- 

# English Auction

- ▶ In the English Auction the auctioneer seeks to find the market price of a good by initially proposing a price below that of the supposed market value and then gradually raising the price.
  - ▶ Each time the price is announced, the auctioneer waits to see if any buyers will signal their willingness to pay the proposed price. As soon as one buyer indicates that it will accept the price, the auctioneer issues a new call for bids with an incremented price.
  - ▶ The auction continues until no buyers are prepared to pay the proposed price, at which point the auction ends. If the last price that was accepted by a buyer exceeds the auctioneer's (privately known) reservation price, the good is sold to that buyer for the agreed price. If the last accepted price is less than the reservation price, the good is not sold
- 

# Example

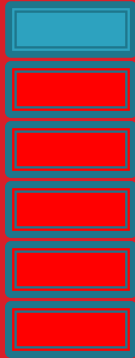
Round 5

Winner

auctioneer

My Cost  
(12)  
15

10




The diagram shows a vertical stack of seven red rectangular blocks. The top block is blue and is currently empty. The second block from the top is red and contains the number 12. The third, fourth, fifth, and sixth blocks are red and contain the number 15. The bottom block is red and contains the number 10. A white arrow with a red outline points from the left towards the stack.

Limit  
Reject  
Accept  
Accepted  
previous

Buyer 1

11



11

The diagram shows a vertical stack of three buttons. The top button is blue and contains the number 11. Below it is a red circular button. Below that is a green circular button. At the bottom is a black rectangular button containing the number 11.

Buyer 2

12




12

The diagram shows a vertical stack of three buttons. The top button is blue and contains the number 12. Below it is a red circular button. Below that is a green circular button. At the bottom is a black rectangular button containing the number 12.

Buyer 3

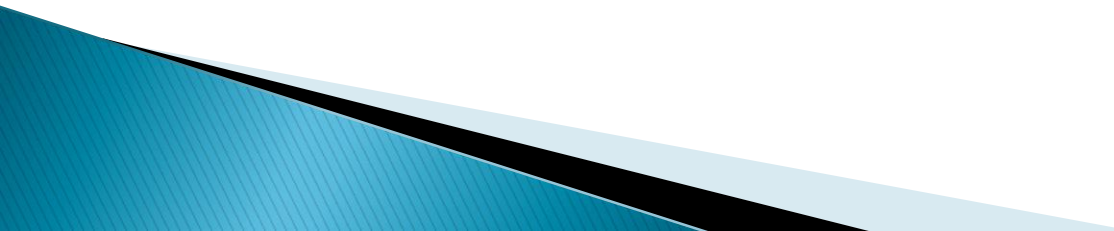
13



13

The diagram shows a vertical stack of three buttons. The top button is blue and contains the number 13. Below it is a red circular button. Below that is a green circular button. At the bottom is a black rectangular button containing the number 13.

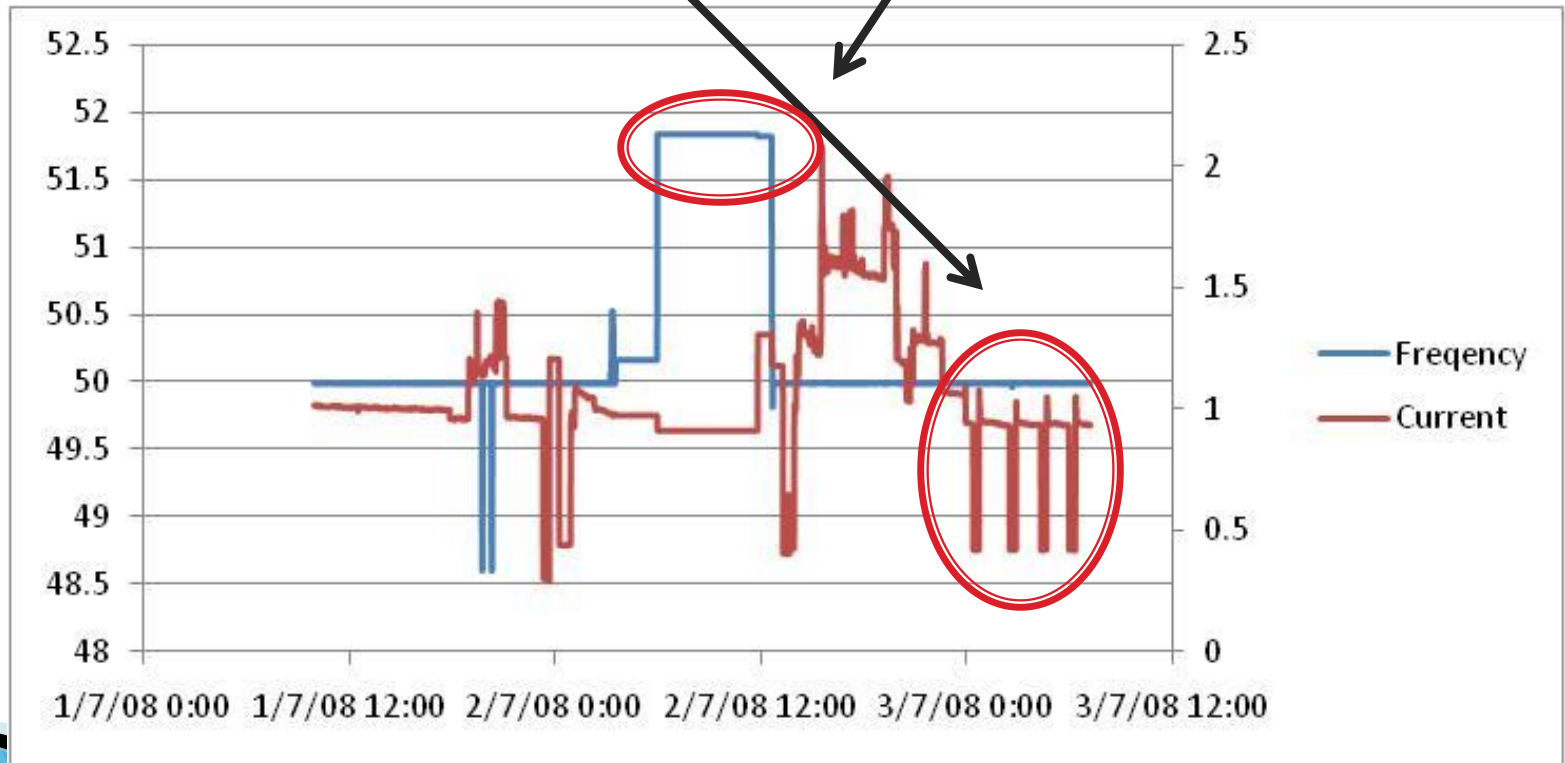
# Example: Policies to estimate the available energy

- ▶ SOC of the Battery: This is an indication of the available energy of the system. The amount of energy above a certain level can be used (example  $> 90\%$ )
  - ▶ The system frequency. If the system frequency is above 50Hz this is an indication that the batteries are full and part of the PV production is rejected
- 

# Example

In this case the frequency is almost 52Hz. This is an indication that the batteries are full and the PV inverters via the droop curves limit their production.

The shedding procedures start later





# Highlights

## Control

- Implementation of Distributed Control
- The houses decide their actions

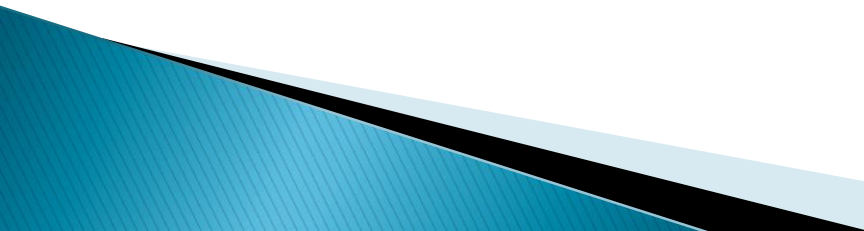
## Software

- Java Based Application
- FIPA Compliant Architecture
- CIM Based ontology

## Hardware

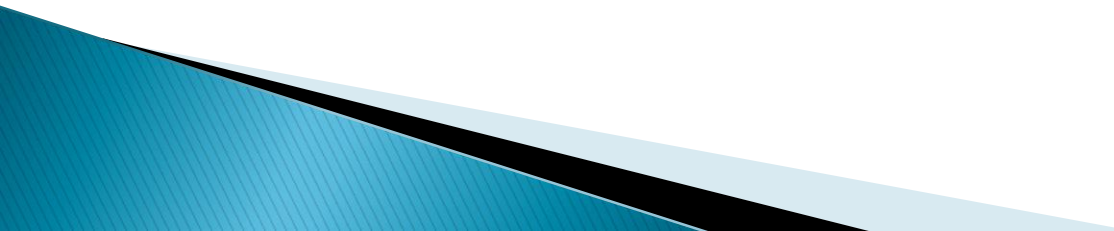
- Embedded Controller
- Communication & Control Capabilities

# Conclusions

- ▶ The Kythnos was the first test site where the MAS system was implemented
  - ▶ A Controller with an Embedded processor has been used to host the agents.
  - ▶ New techniques have been tested such as: negotiation algorithms, wireless communication, CIM based ontology etc..
  - ▶ The architecture is too complex for small systems but offers great scalability.
- 

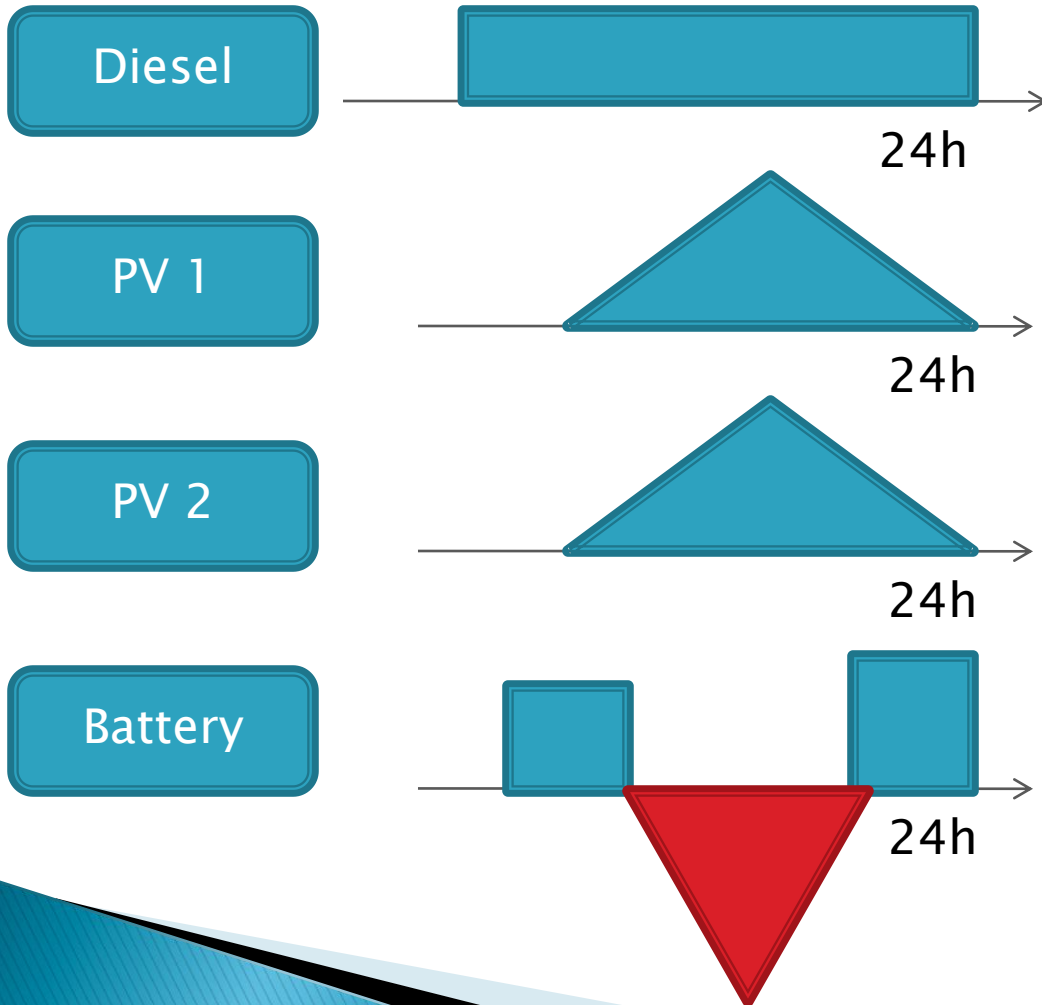
## Example #3

# Scheduling using MAS Technology

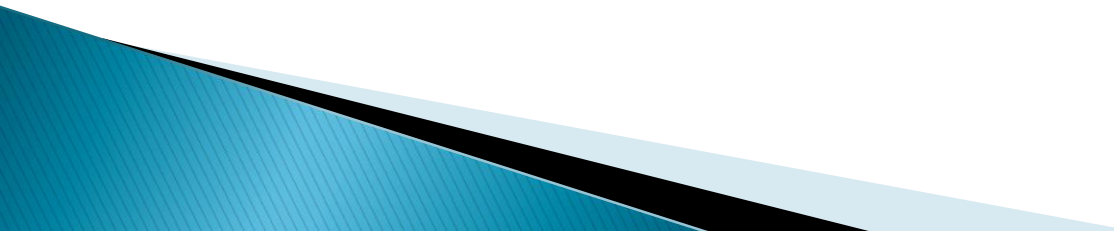
- ▶ This is a scheduling problem: the agent should decide a set of actions in the next period (example 24h).
  - ▶ The next problem tested here is the Scalability
  - ▶ The system was tested in a real distributed test site in Athens
- 

# The problem

The goal is to combine several units in order to act as a single unit

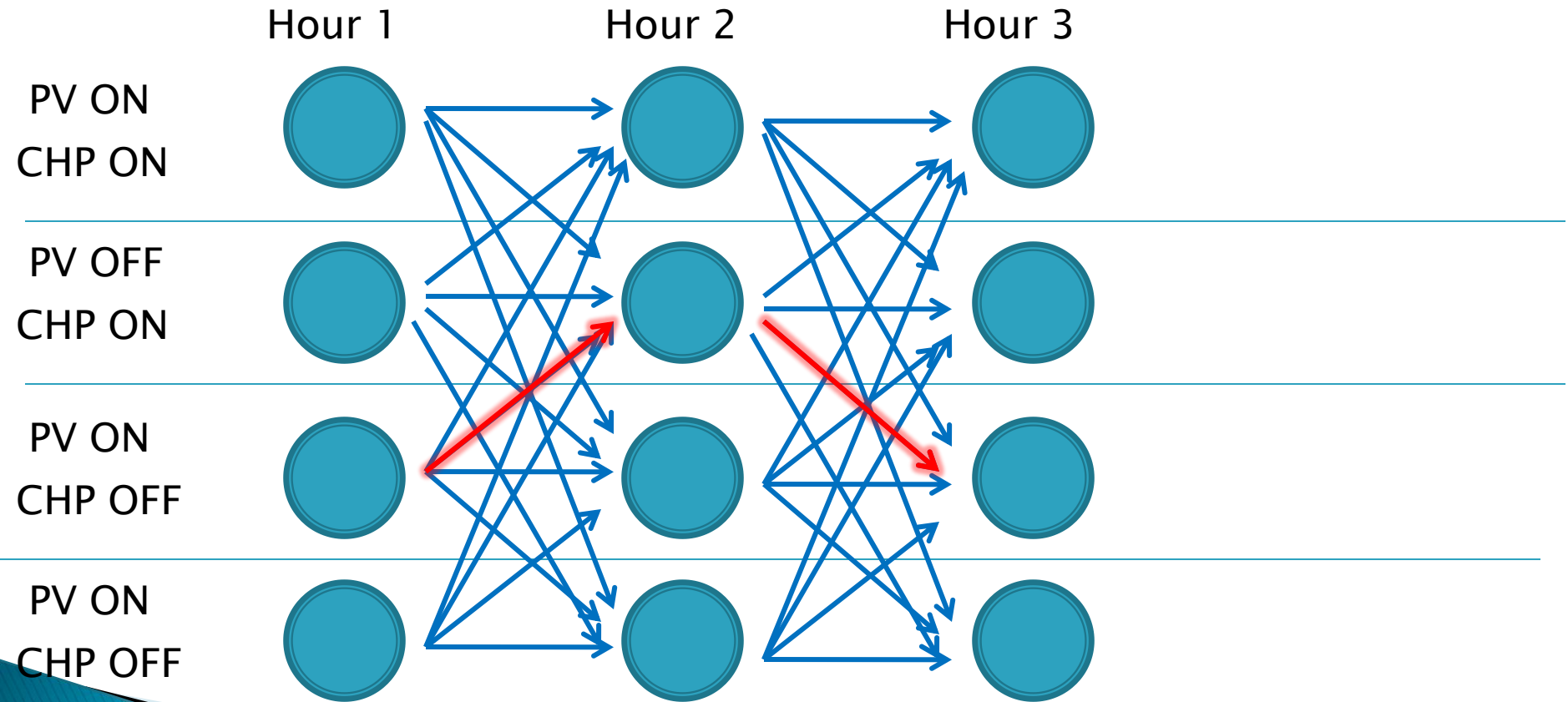


# The theoretic problem

- ▶ The agents should learn how to create a schedule of actions
  - ▶ The first problem is that the environment is stochastic: for example you cannot predict the production of a PV
  - ▶ The second problem is that the system should reduce the amount of data exchange: knowing the decisions of all agents includes a lot of information.
- 

# Reinforcement Learning Algorithm

A graph represents the system states during transitions of an office system.  
The scheduling problem is equivalent with the problem of finding the shortest path.  
Its one has a weight representing the cost of this transition

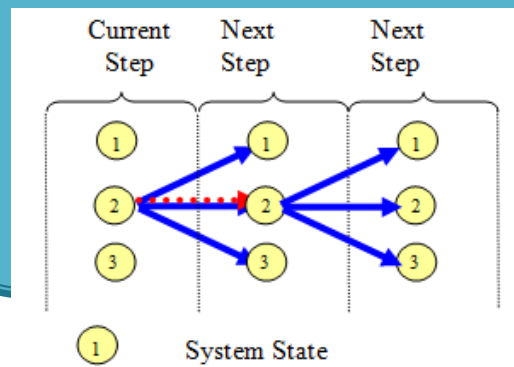


# Multi Agent Reinforcement Learning

## Multi Agent Reinforcement Learning

1. State  $s$
2. Action  $a$
3. Immediate Reward  $r$

$$Q(s, a) = (1 - k)Q(s, a) + k(r + \gamma \max_{a'} Q(s', a'))$$

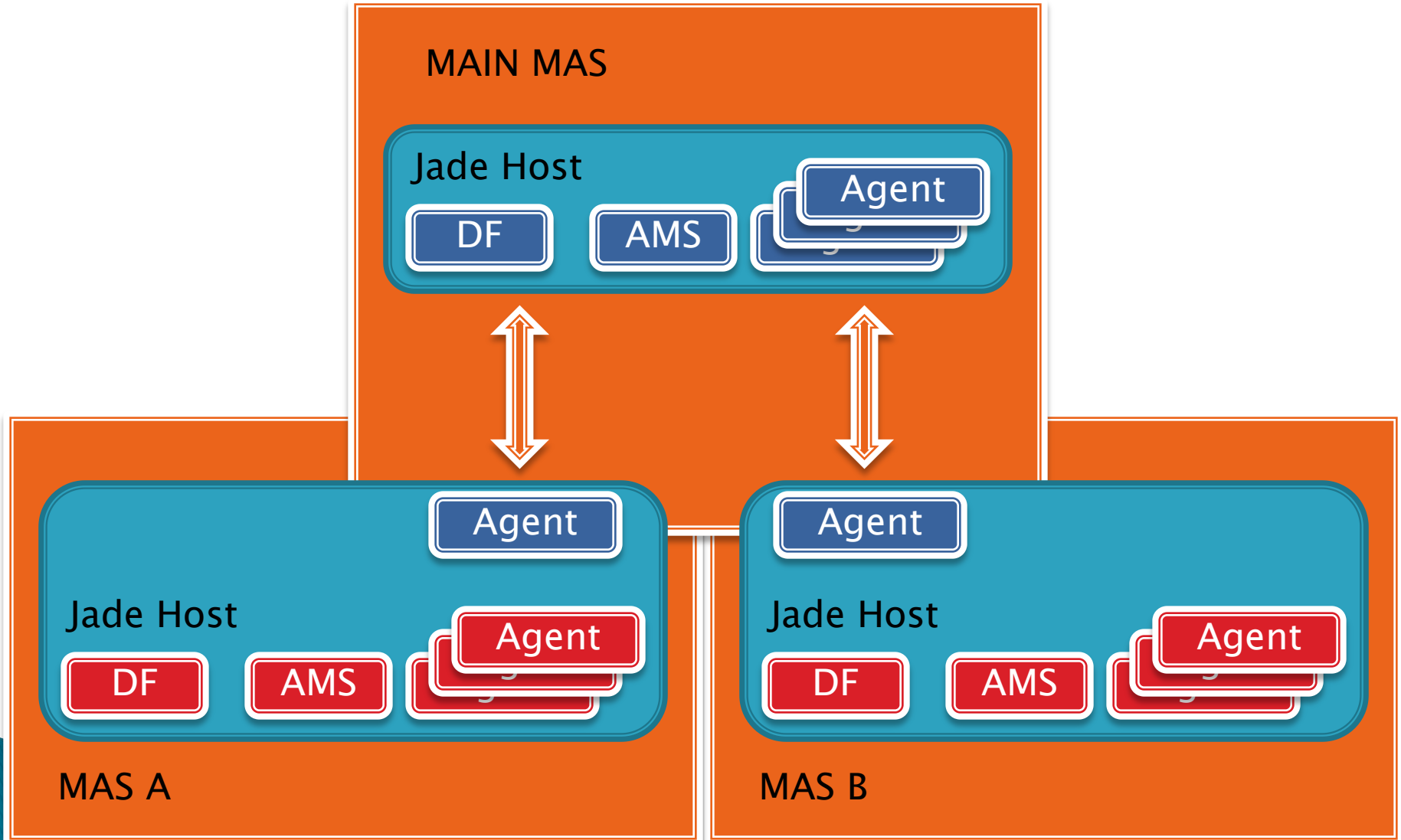


Training: The system creates the Q Matrix which has the information of what the best action is.




Operation: The agent make decision according to the Q Matrix

# Implementation



# The implemented example in the laboratory

- ▶ Black Start support
  - ▶ The agents learn to start the system and keep it running for the longest possible period in island mode.
  
  - ▶ Sequence of actions:
    1. Switch off all loads
    2. Launch black start units
    3. Launch the other units
    4. Start the MAS according to the results of the Reinforcement Learning.
- 

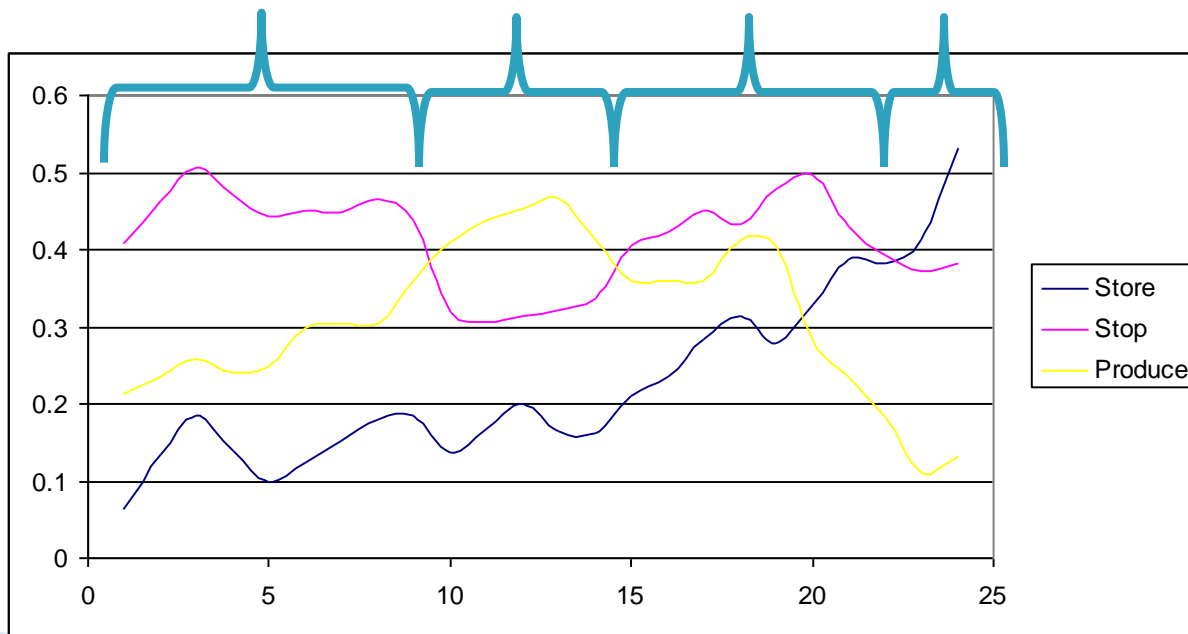
# Types of actions

	Type of agent	Actions
1	Load	On
		Off
2	Storage Unit	Produce
		Stop
		Store
3	Production Unit	Stop
		Produce

# Results of the algorithm

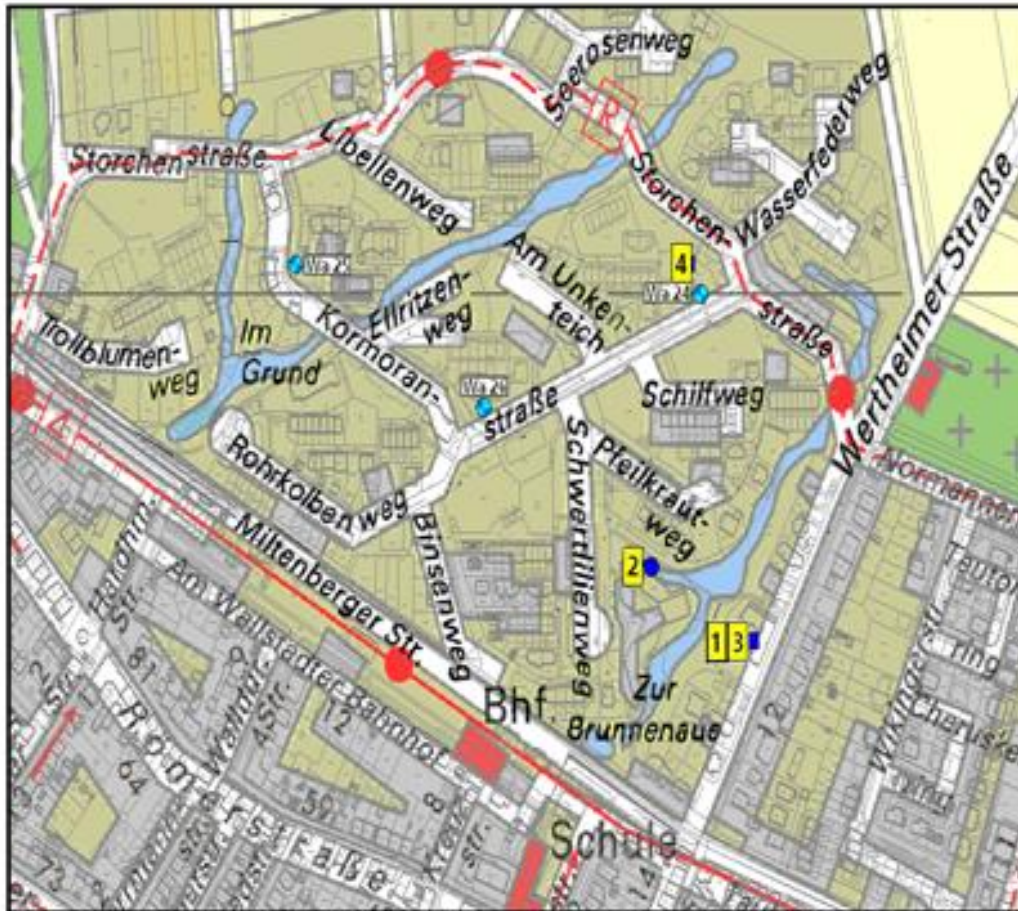
This is part of the Q table of the battery. It provides information about the best action according to the system state.

Do nothing    Produce    Do nothing    Store



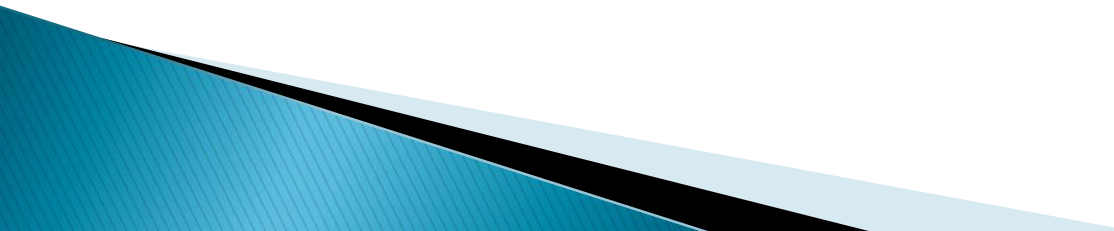
# Example #4

## Wallstadt installation

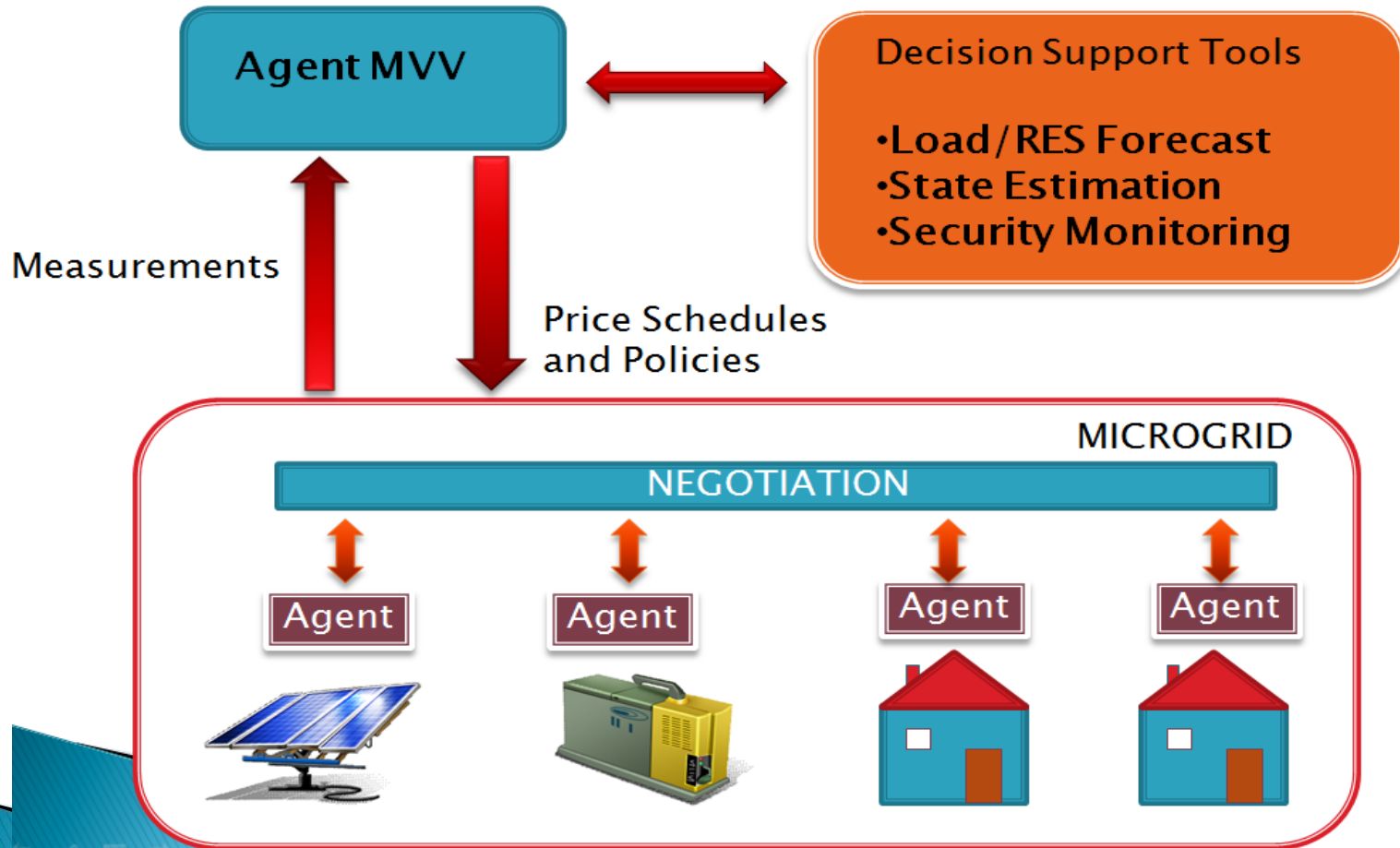


- 1** PV System 1  
14,28 KWp
- PV System 2  
9,18 KWp
- 2** Pump  
3 x 0,75 KW
- 3** Kinderhaus  
Convactor heater: 10,3 KW
- 4** Battery  
60 KW in 3 hours

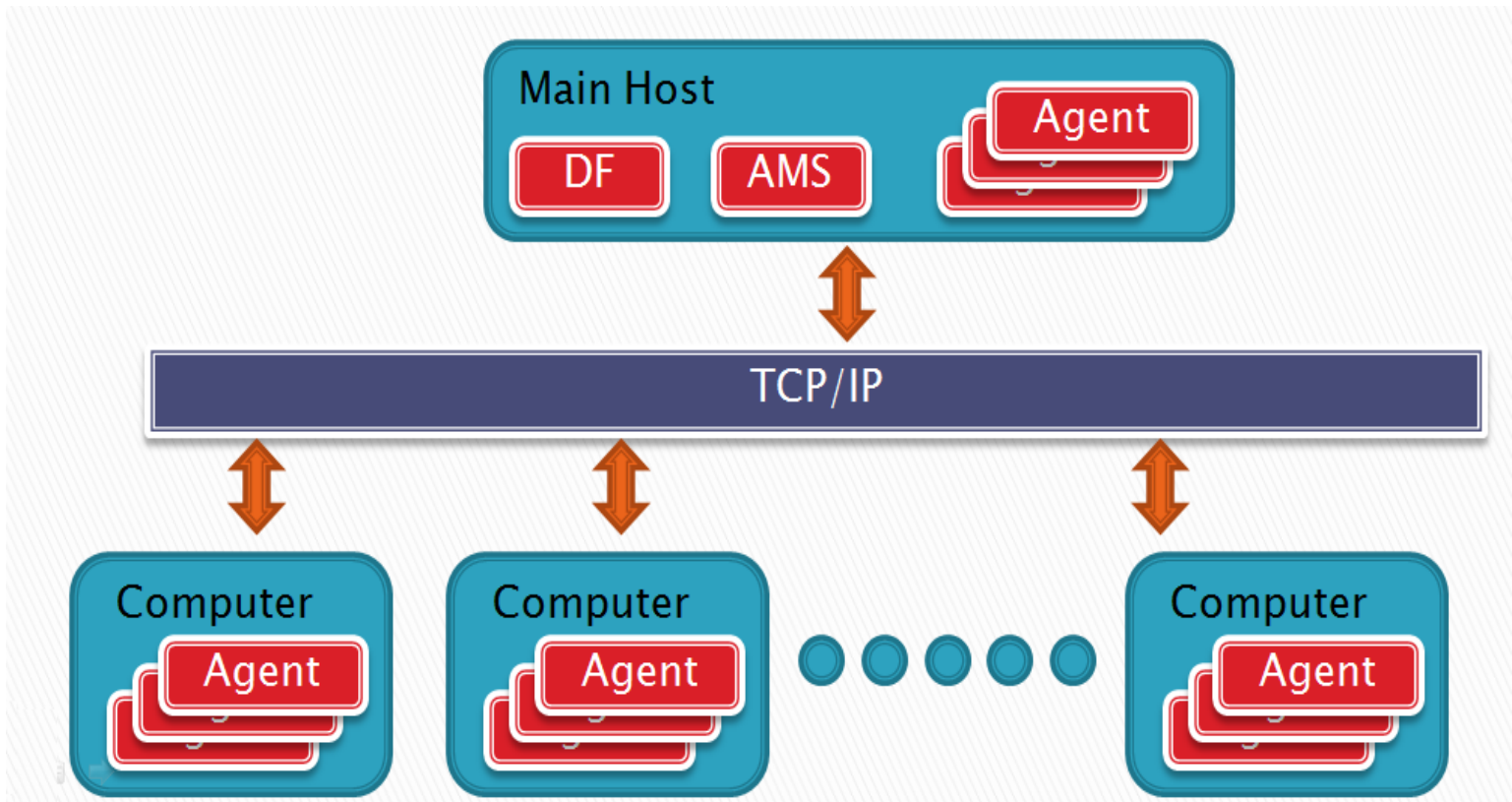
# Goal of the experiment

- ▶ The goal of the experiment is to test the transition to a virtual island state.
  - ▶ The consumption should be equal to the
- 

# The configuration of the test site

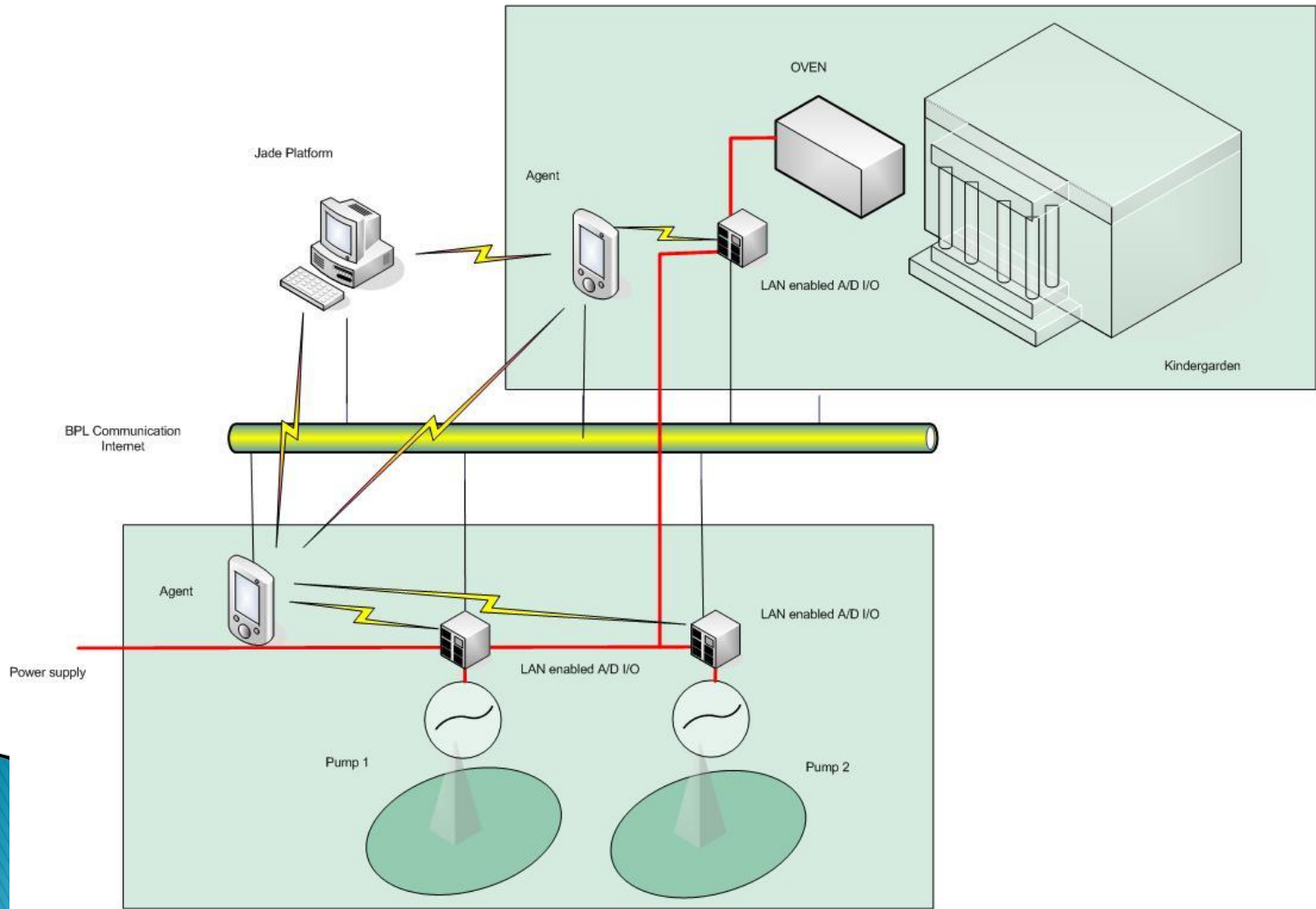


# The configuration of the agent software



# The physical configuration

The agent run on a remote PC and communicate via LAN with the Distribute I/O modules



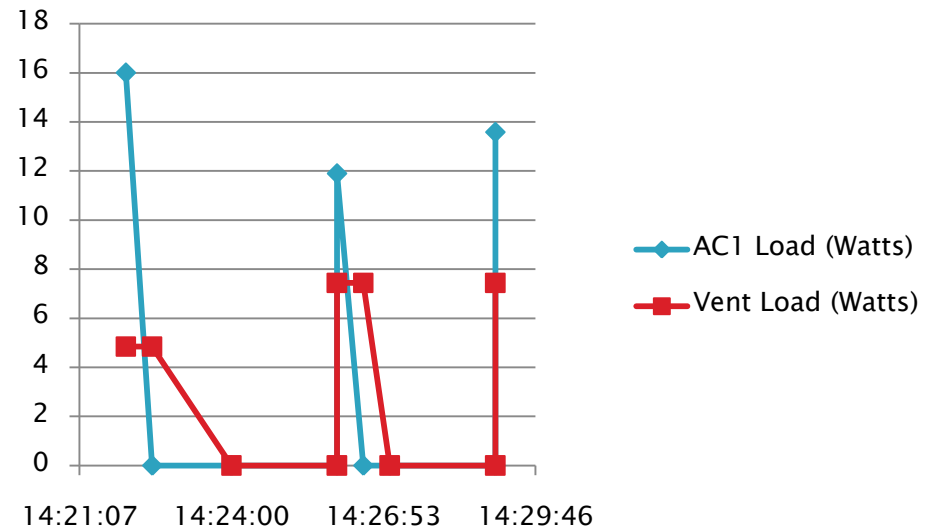
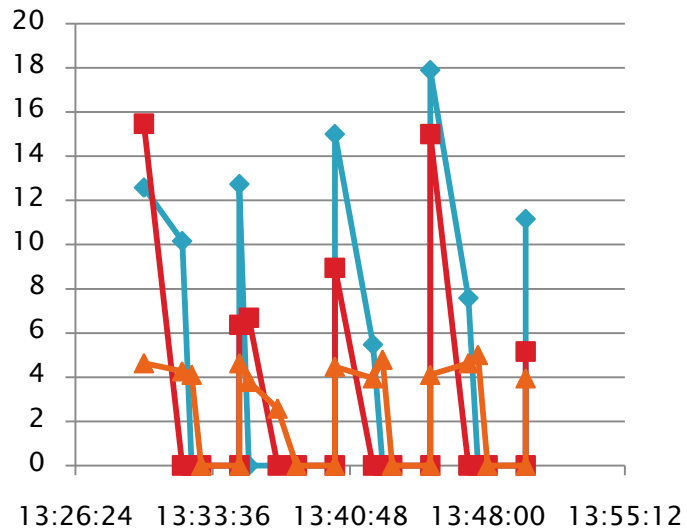
# Screenshot of the Jade

The screenshot displays the Jade development environment, which includes several windows and a taskbar.

- RedAnaLab - JCreator:** The main IDE window, showing a code editor with a C++ program. The code includes comments and logic for load shedding, such as "A Load has Been Shedded.MGCC takes no action. Just waits." and "The number of agents is: 4".
- Command Prompt (C:\WINDOWS\system32\cmd.exe):** Multiple instances showing the execution of the program. The output includes CIM messages, MGCC received messages, and negotiation status updates. For example, "WallstadtAC01@192.168.2.199:1099/JADE -> Command Received for a NEW Negotiation Cycle" and "I can shed 11.897459 W for the next Negotiation Cycle".
- JADE Remote Agent Management GUI:** A window titled "rma@192.168.2.199:1099/JADE - JADE Remote Agent Management GUI". It features a tree view under "AgentPlatforms" showing a hierarchy of agents: "192.168.2.199:1099/JADE" (Main-Container) containing "rma@192.168.2.199:1099/JADE", "df@192.168.2.199:1099/JADE", and "ams@192.168.2.199:1099/JADE". Below these are several "BE-192.168.2.199\_1099-43" through "45" containers, each containing a "Wallstadt" agent (e.g., "WallstadtAC01@192.168.2.199:1099/JADE", "WallstadtVent@192.168.2.199:1099/JADE", "WallstadtMainAg@192.168.2.199:1099/JADE").
- Taskbar:** Shows the Windows taskbar with the Start button, taskbar buttons for "RedAna...", "Jade", "rma@19...", "RedAna...", "Advante...", and several "C:\WIN..." windows. The system tray shows the time as "Ln 150 Col 77 Char 77 OVR Res".

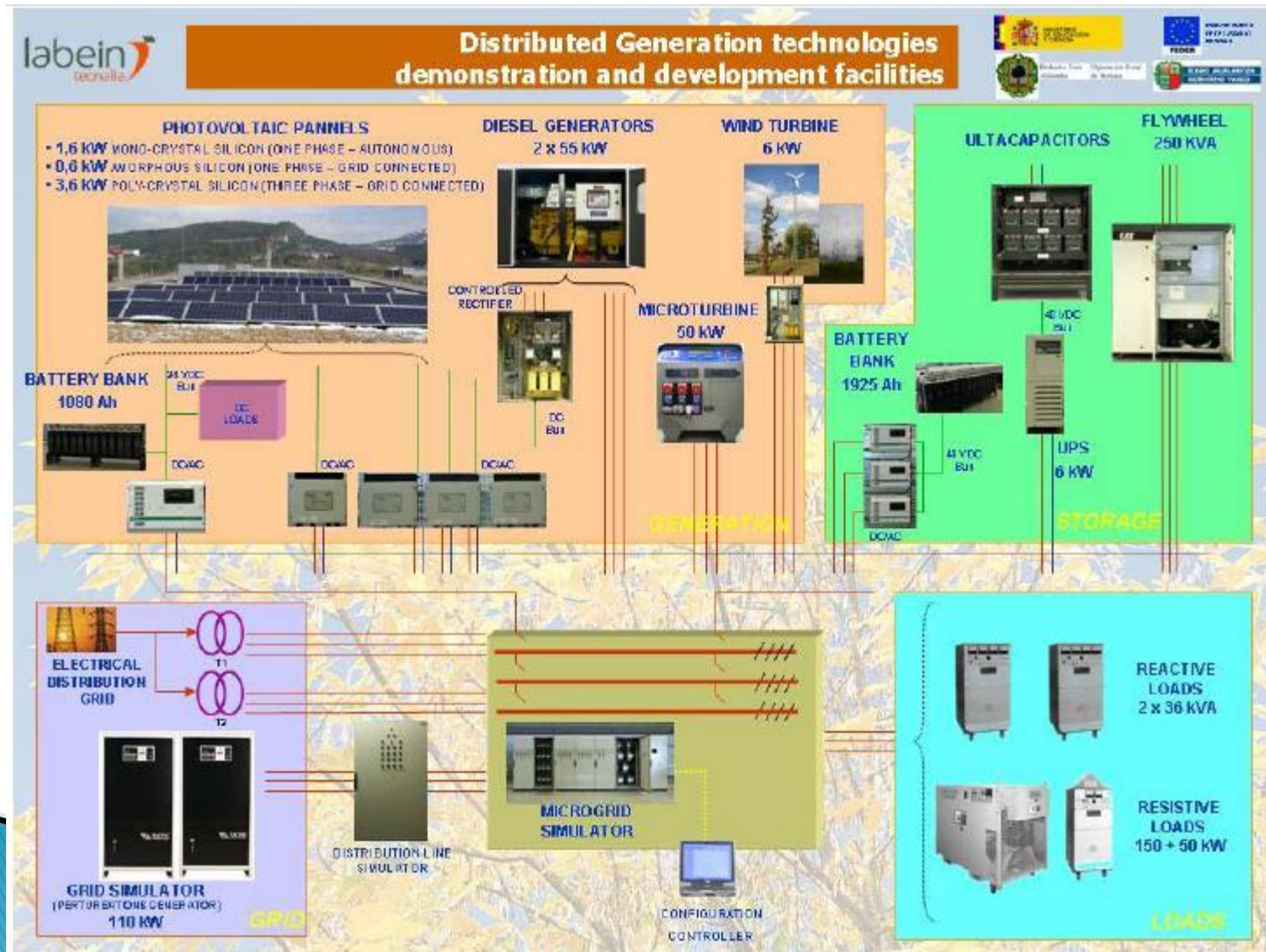
# Results

Load shedding schedule from the agents

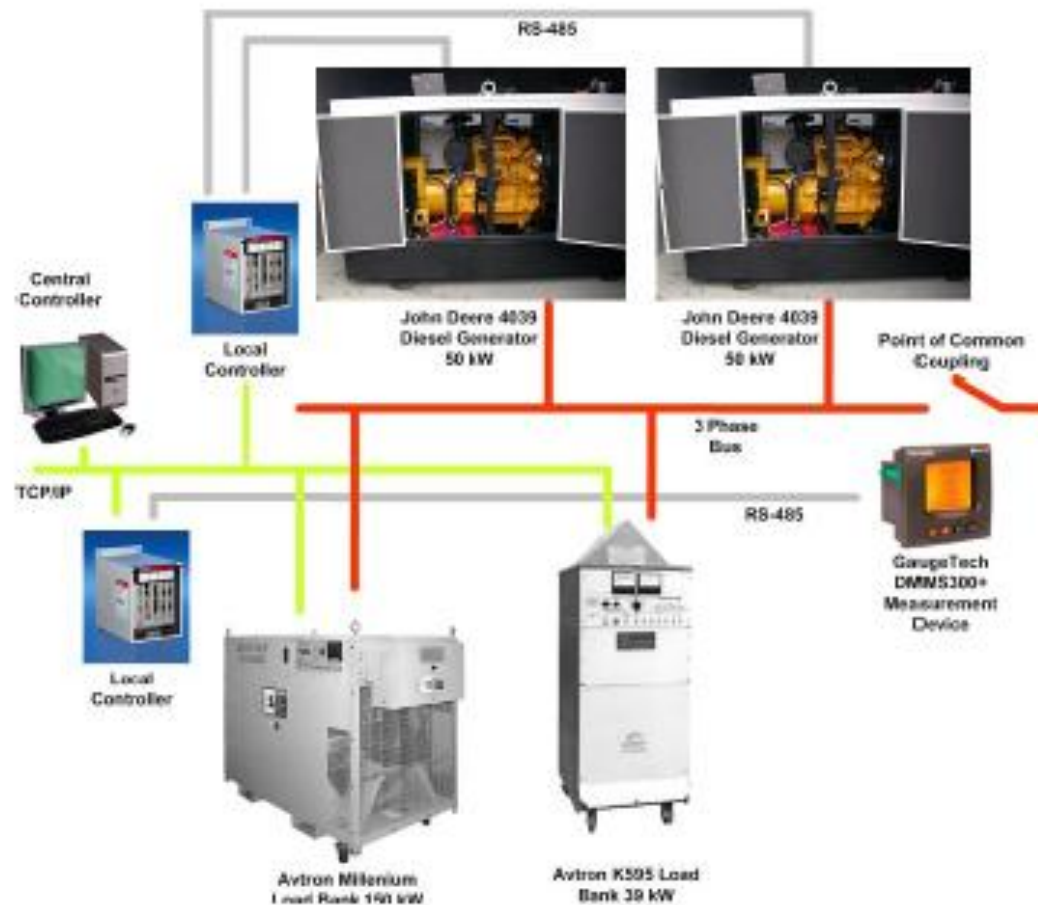


# Example #5

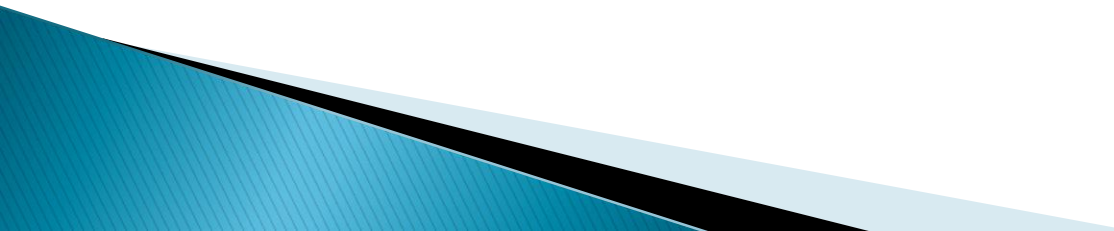
## Installation in LABEIN



# Laboratory Overview



# The main characteristics of the developed system

- ▶ Integration of legacy systems
  - ▶ Plug & Play capability
  - ▶ Extensible
- 

# Secondary Frequency Control

The objectives of the secondary regulation are:

- ▶ Microgrid in grid connected mode: In this case the secondary regulation will try to maintain a previously defined power exchange schedule with the main grid.
- ▶ Microgrid in islanded mode: When the microgrid is in islanded mode, the role of the secondary regulation control will be to maintain the frequency in the microgrid as close as possible to a reference frequency.

# Results

