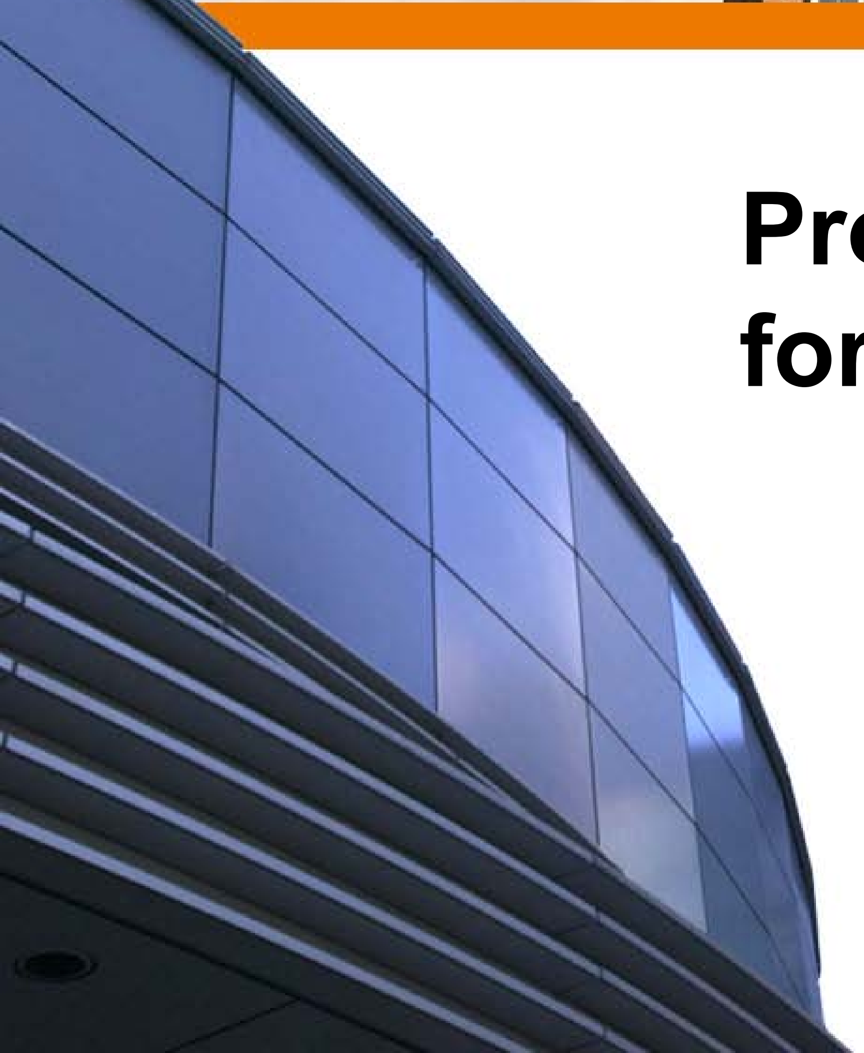


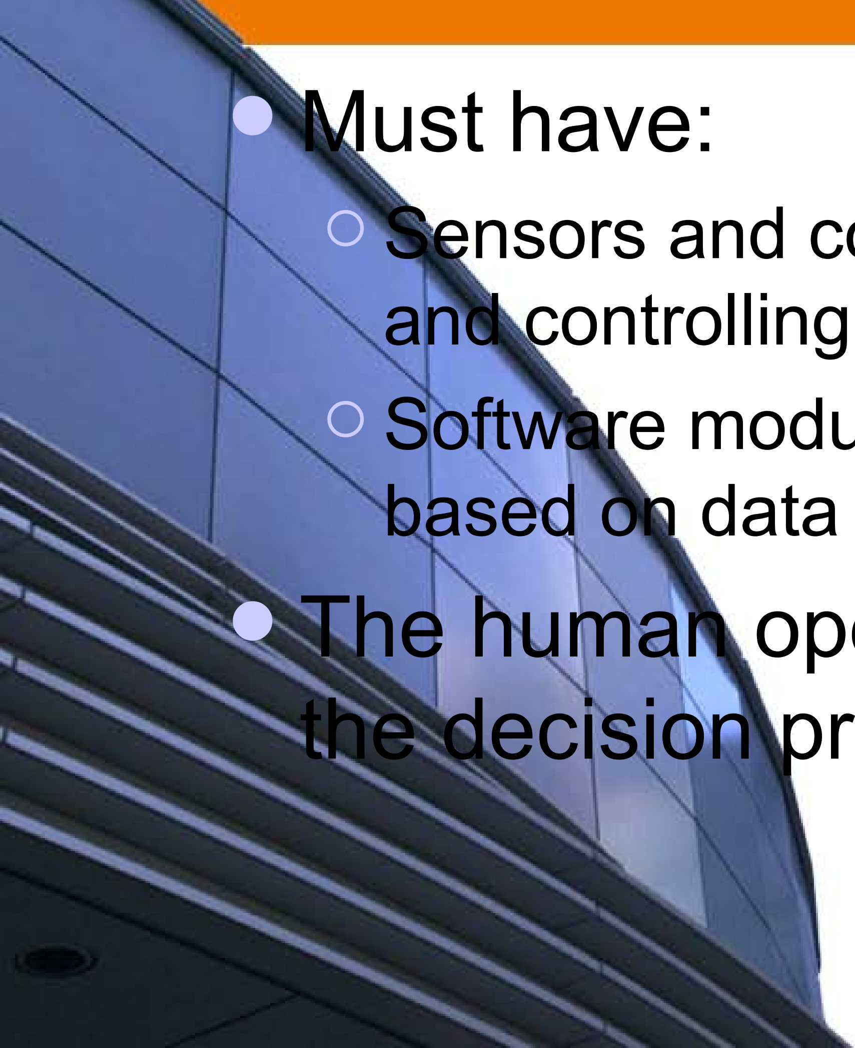
Programming manual for smart houses

Genge Bela

“Petru Maior” University, Tg. Mures

consultant Intelligent Building Solutions

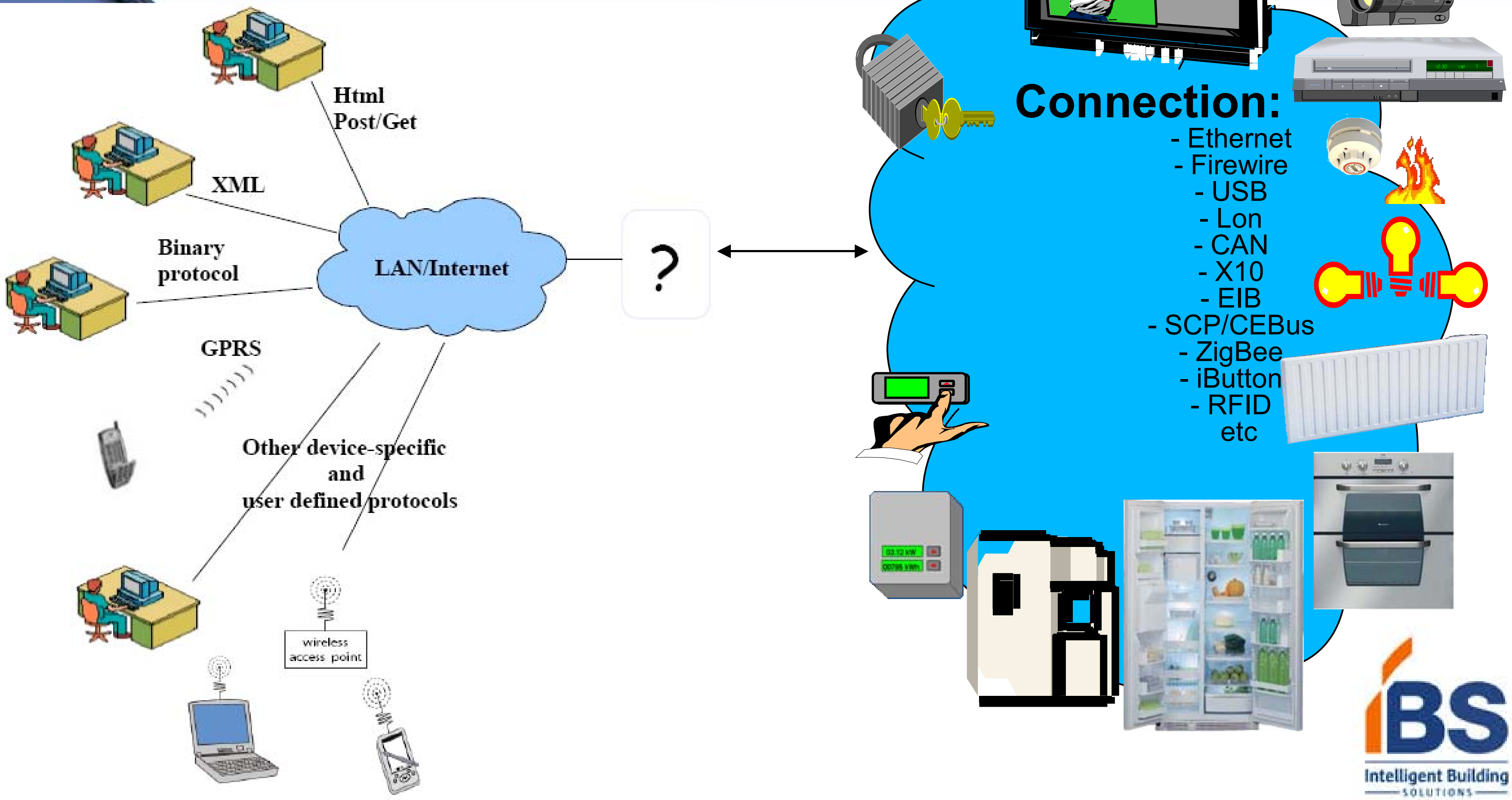
What is a smart house?



- Must have:
  - Sensors and control elements for monitoring and controlling physical equipments
  - Software modules that can take decisions based on data received from sensors
- The human operator should not interfere in the decision process

iBS
Intelligent Building
SOLUTIONS

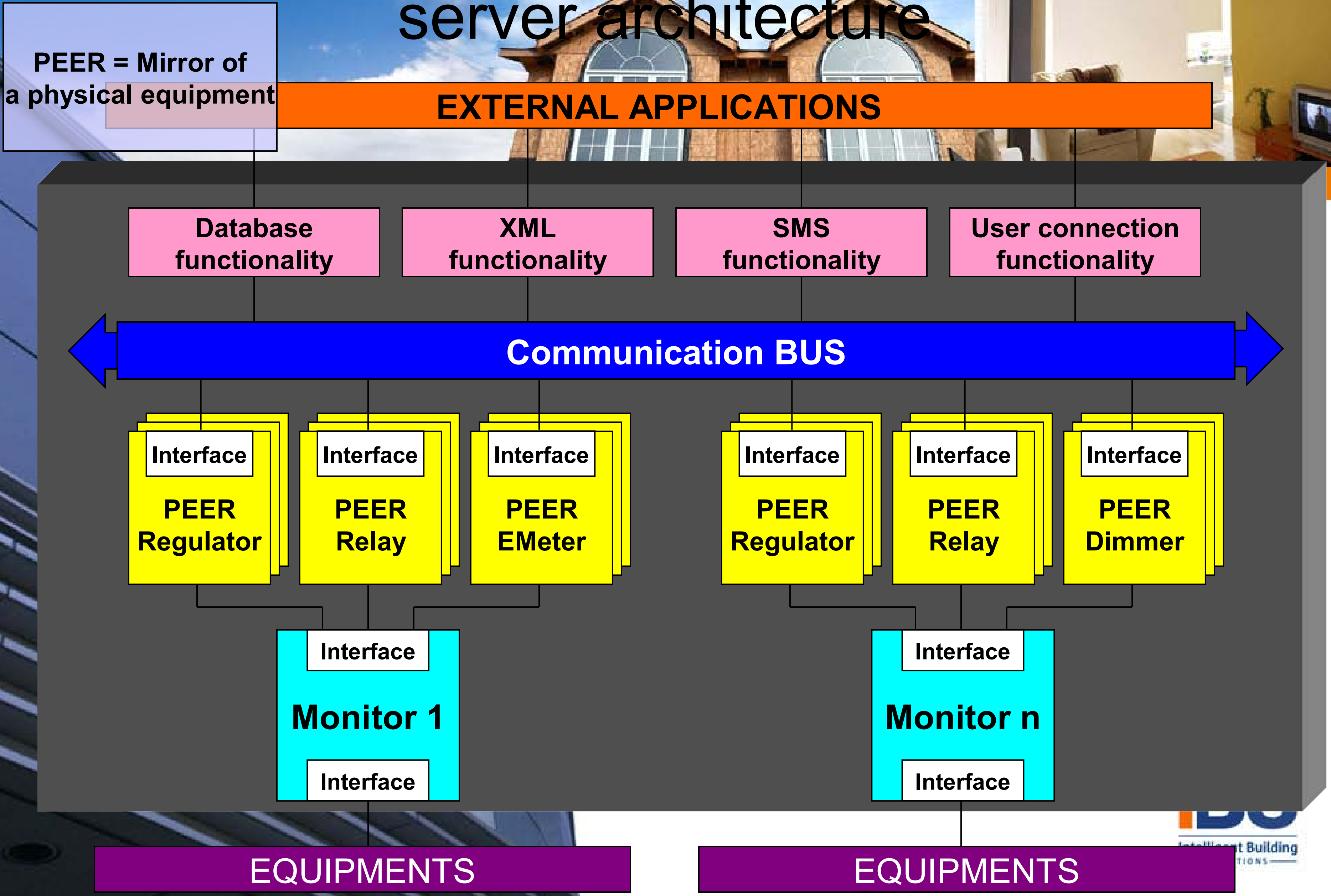
The main problem: multi-protocols for controlling diversed equipments



Connection:
- Ethernet
- Firewire
- USB
- Lon
- CAN
- X10
- EIB
- SCP/CEBus
- ZigBee
- iButton
- RFID
etc

IBS Intelligent Building Solutions

ESS ("Environmental Service System") server architecture



PEER = Mirror of a physical equipment

EXTERNAL APPLICATIONS

Database functionality

XML functionality

SMS functionality

User connection functionality

Communication BUS

Interface
PEER Regulator

Interface
PEER Relay

Interface
PEER EMeter

Interface
PEER Regulator

Interface
PEER Relay

Interface
PEER Dimmer

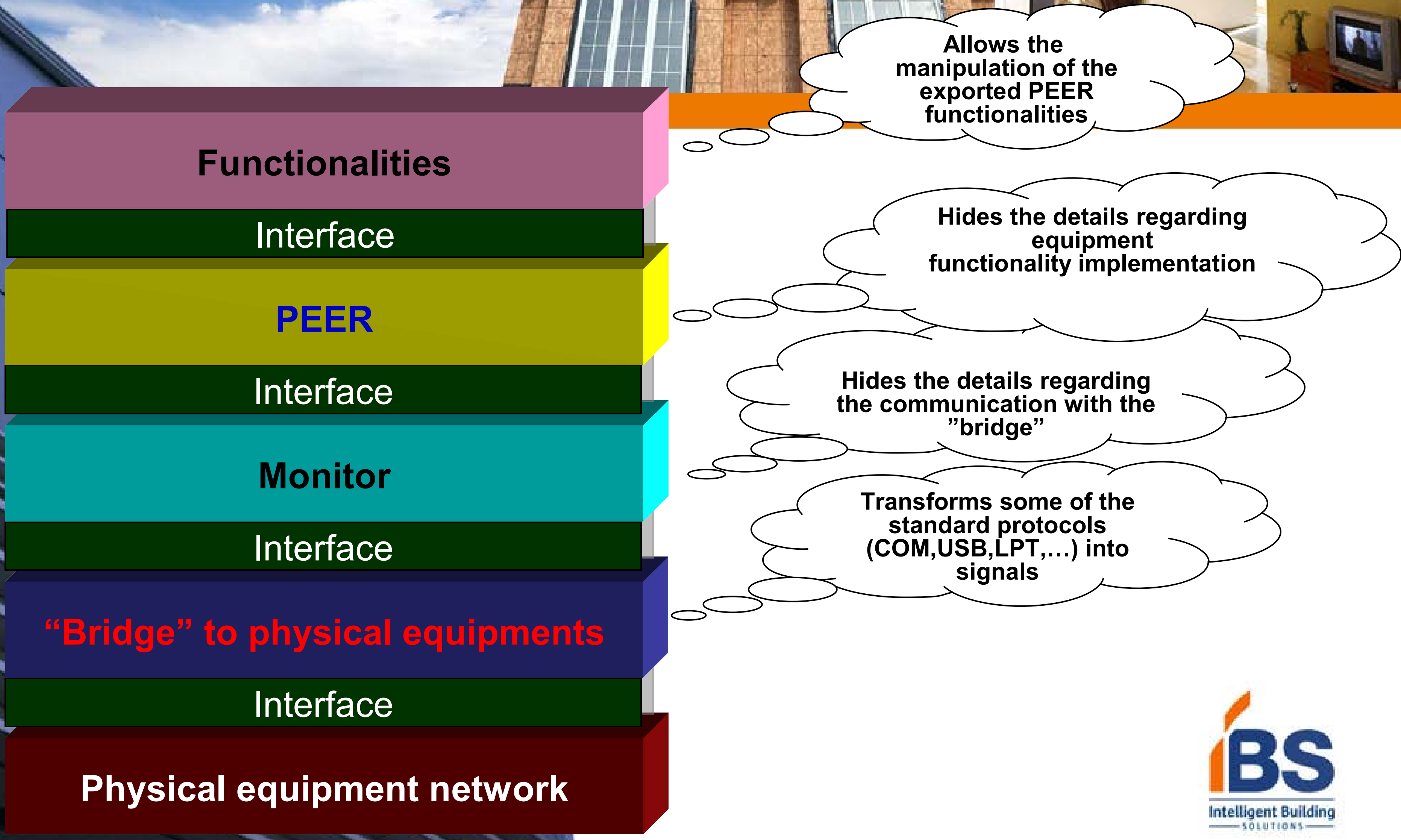
Interface
Monitor 1
Interface

Interface
Monitor n
Interface

EQUIPMENTS

EQUIPMENTS

Levels of abstractisations



Functionalities

Interface

PEER

Interface

Monitor

Interface

“Bridge” to physical equipments

Interface

Physical equipment network

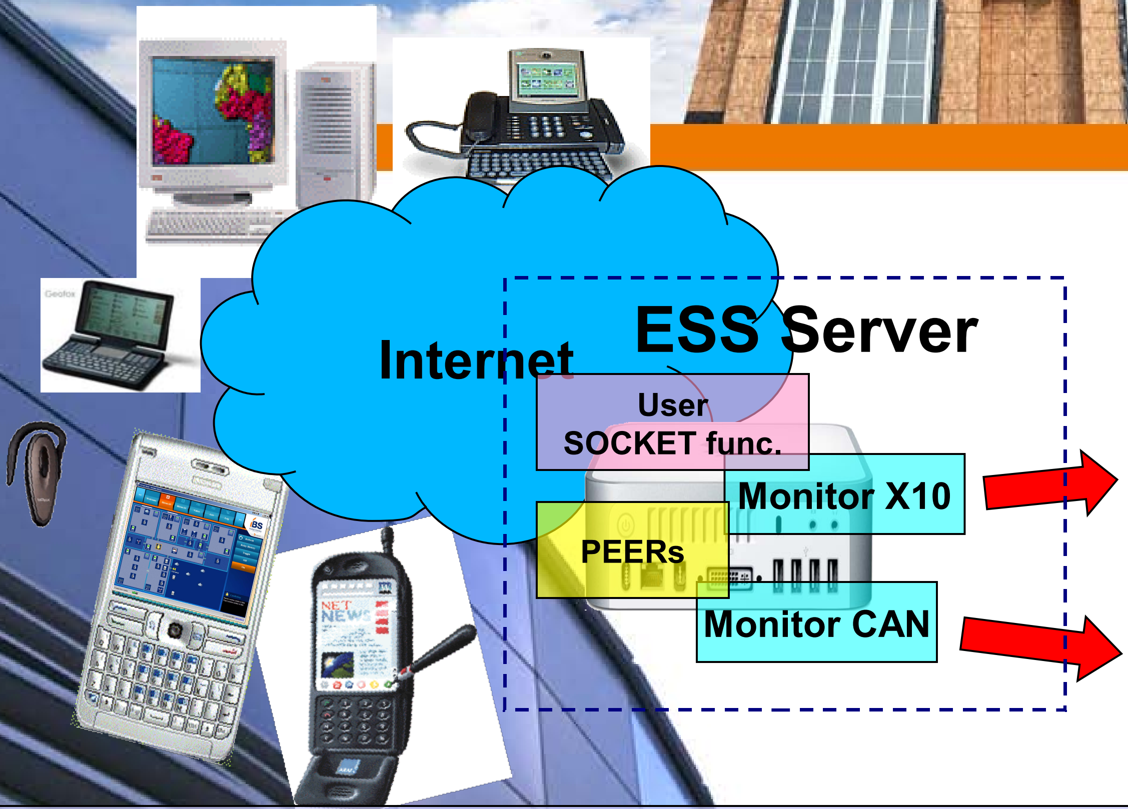
Allows the manipulation of the exported PEER functionalities

Hides the details regarding equipment functionality implementation

Hides the details regarding the communication with the ”bridge”

Transforms some of the standard protocols (COM,USB,LPT,…) into signals

ESS = Environmental Service System
- Extensibility according to the controlled env. -



Internet

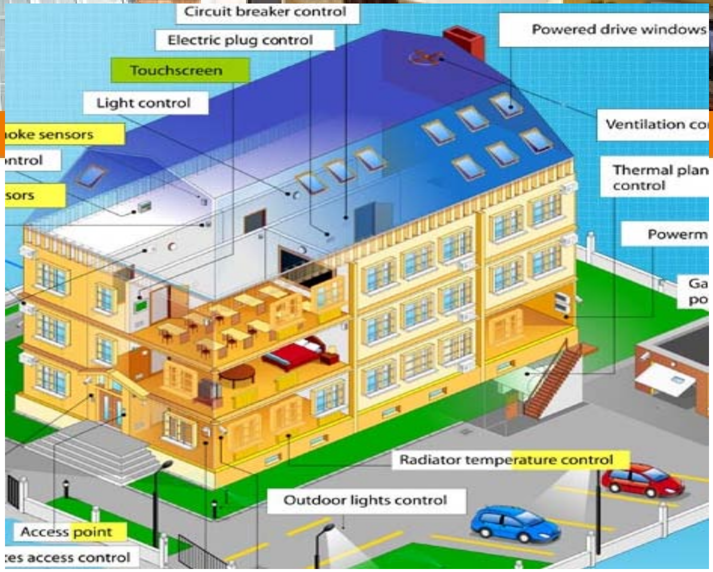
ESS Server

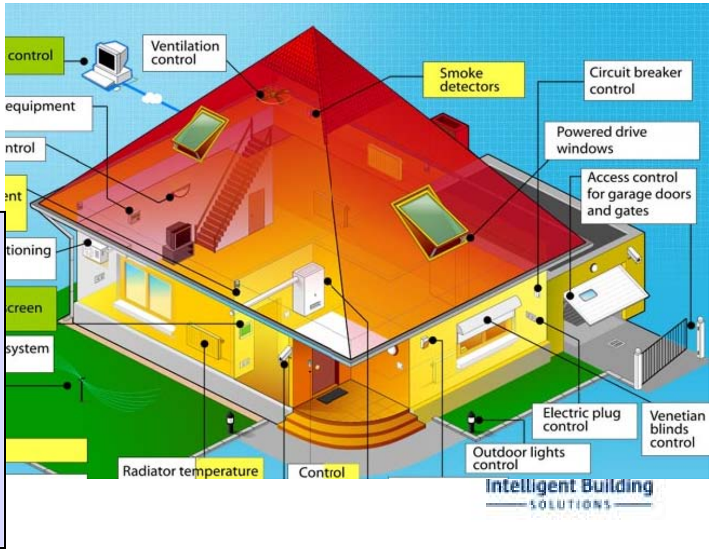
User SOCKET func.

Monitor X10

PEERs

Monitor CAN

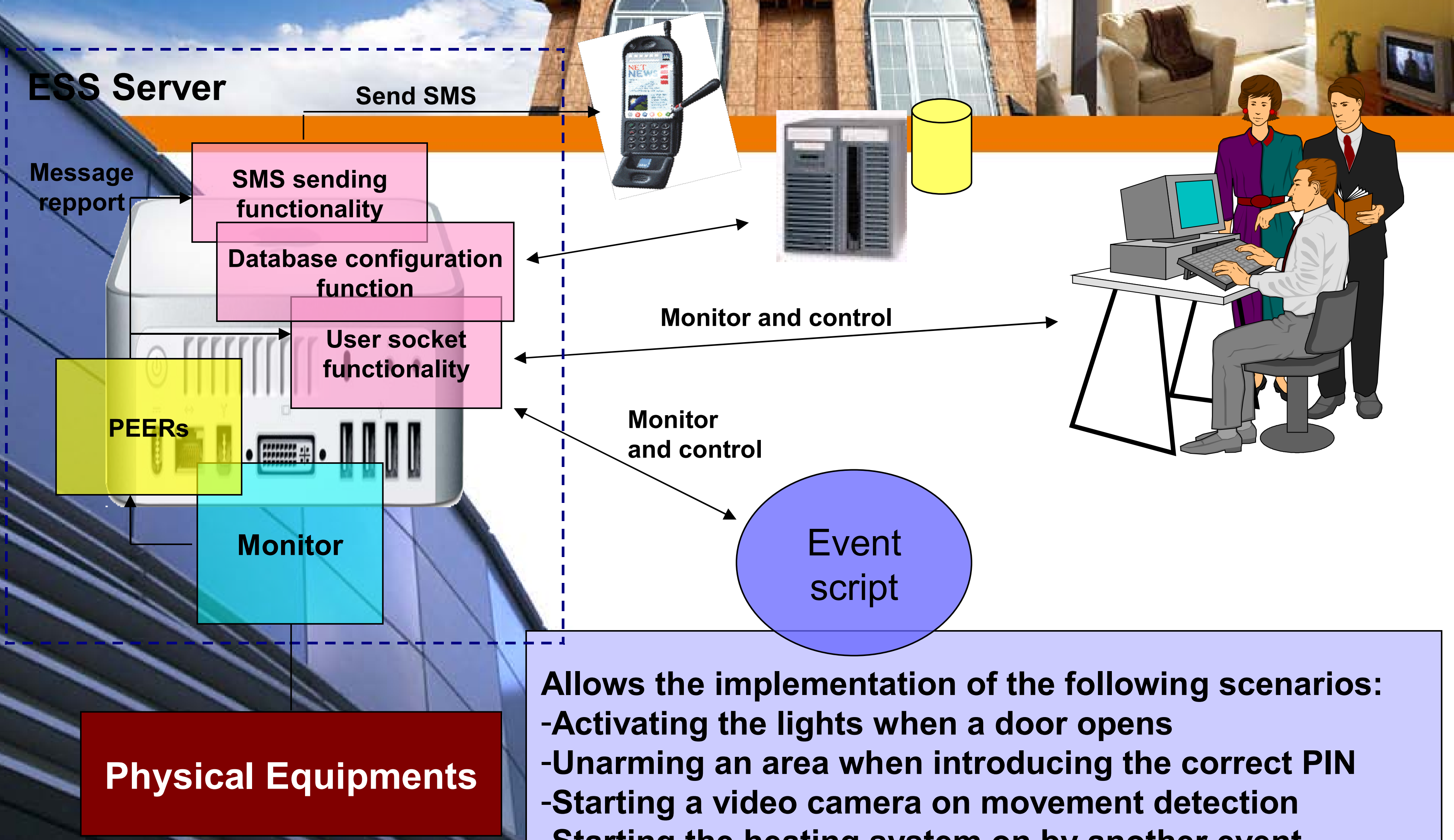




X10 – International communication standard based on electrical power lines

CAN – "Controller Area Network", international communication standard based on a BUS similar to Ethernet

ESS = Environmental Service System
- Extensibility according to the user's demands -



ESS Server

Send SMS

Message repport

SMS sending functionality

Database configuration function

User socket functionality

PEERs

Monitor

Monitor and control

Monitor and control

Event script

Physical Equipments

Allows the implementation of the following scenarios:
-Activating the lights when a door opens
-Unarming an area when introducing the correct PIN
-Starting a video camera on movement detection
-Starting the heating system on by another event

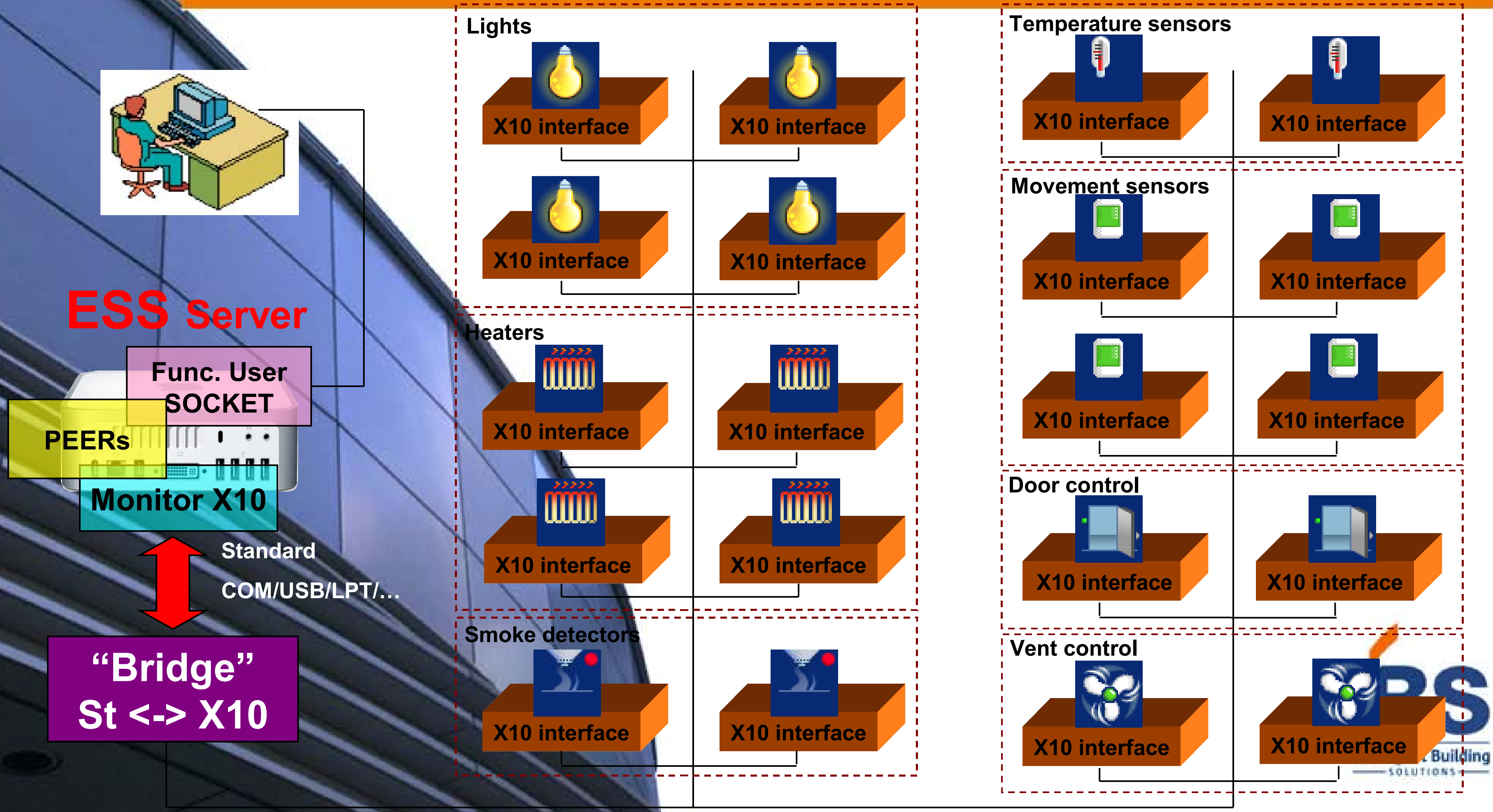
Easy to use protocol

- The architecture allows a simple and clear replacement of the basic user protocol module
- The implemented protocol was built on simplicity and clarity:
  - Request-response protocol
  - Parameter separator that is easy to parse
- Message examples:
  - User->ESS: (**GET**)(**PEERS**)(*demon*)(*monitor*)
  - ESS->User: (**NEW**)(**PEERS**)(*demon*)(*monitor*)(*peer1*)(*peer2*)...(*peerN*)

System example

Monitoring and controlling equipments connected via X10

X10 – International communication standard based on electrical power lines



ESS Server

Func. User SOCKET

PEERs

Monitor X10

Standard

COM/USB/LPT/…

“Bridge”
St <-> X10

Lights

X10 interface

X10 interface

X10 interface

X10 interface

Heaters

X10 interface

X10 interface

X10 interface

X10 interface

Smoke detectors

X10 interface

X10 interface

Temperature sensors

X10 interface

X10 interface

Movement sensors

X10 interface

X10 interface

X10 interface

X10 interface

Door control

X10 interface

X10 interface

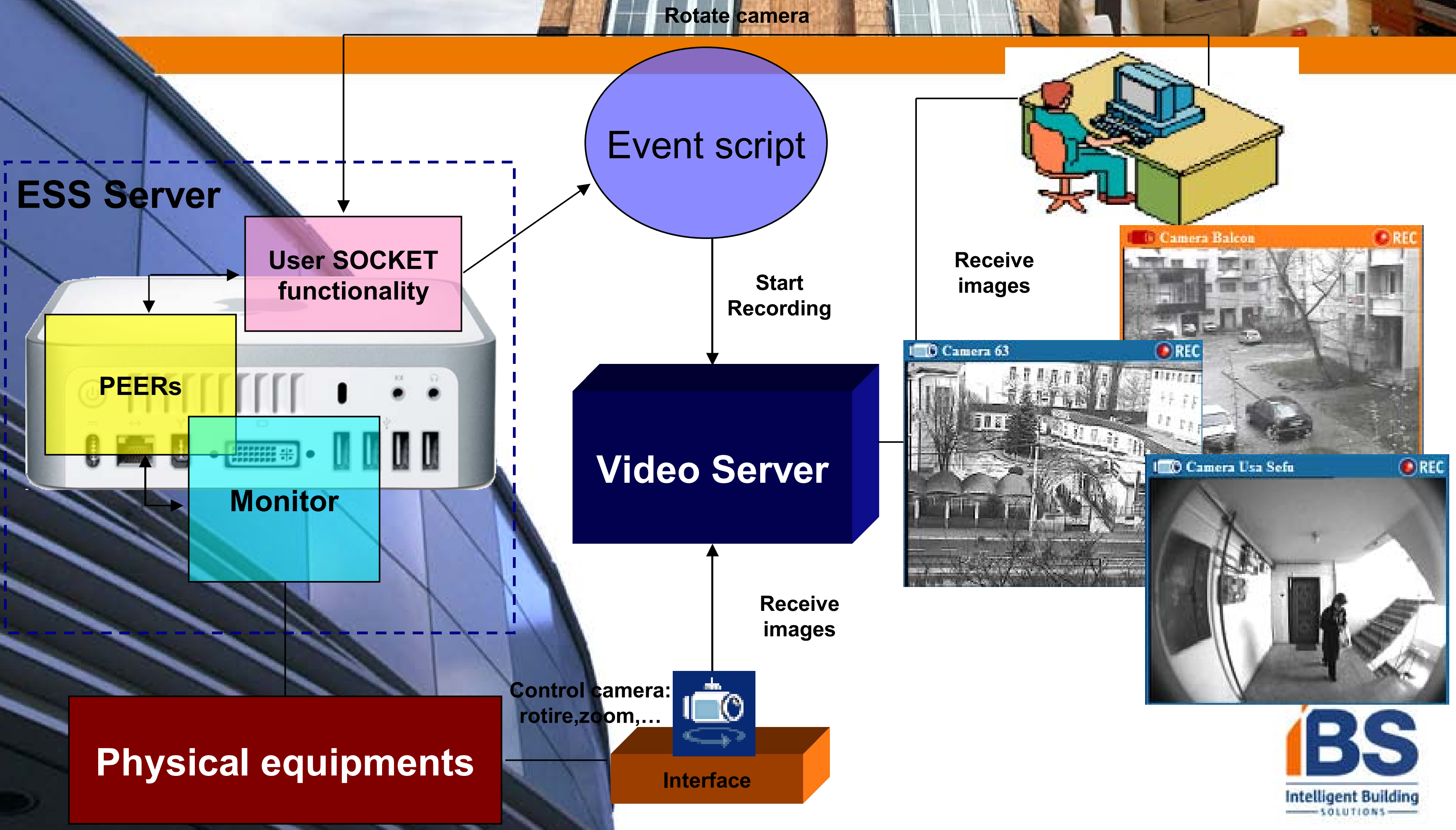
Vent control

X10 interface

X10 interface

System example

Video cammeras controlled by movement



Rotate camera

Event script

ESS Server

User SOCKET functionality

PEERs

Monitor

Start Recording

Video Server

Receive images

Receive images

Control camera: rotire,zoom,…

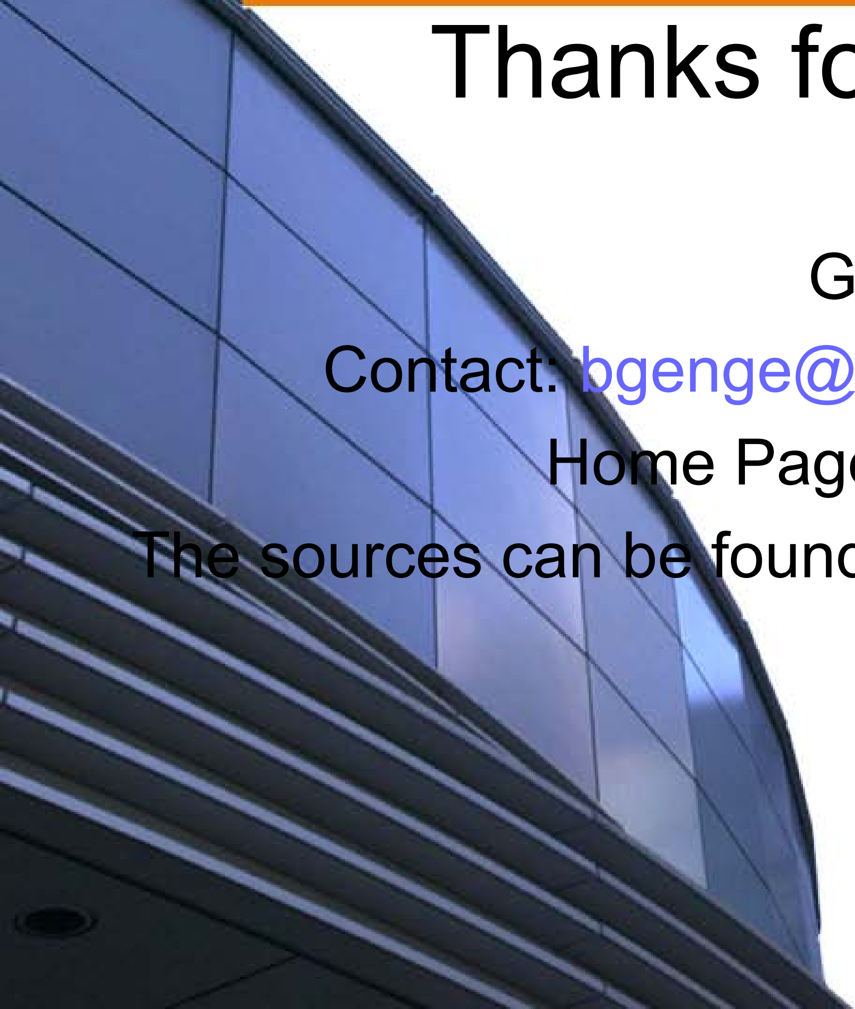
Interface

Physical equipments

Moving towards the Open Source community

- Opened sources:
  - ESS server
  - X10 equipment software and schematics
- Benefits to developers:
  - Integrate own type of equipments
  - Develop new scenarios
  - Develop new functionality modules (e.g. GPS) or hardware modules (e.g. CEBus)
  - Develop a monitoring system using practically any communication protocol
- Perspectives on the future:
  - Open the code for other integrated applications: Video, Replay, XML
  - Open the code for client applications: Standalone monitoring client, Mozilla XUL-based monitoring client

Thanks for the attention!



Genge Bela

Contact: bgenge@upm.ro, bela.genge@ibs.ro

Home Page: www.ibs.ro/~bela

The sources can be found at: http://www.ibs.ro/opensource