leds chat

5th International LED professional Symposium +Expo
LpS *2015* | Sept 22-24, 2015 | Bregenz

LED's CHAT
a new software approach/architecture for scalable distributed lighting control

Peter Niebert
Aix-Marseille University / LED's CHAT

LpS 2015
LED professional SYMPOSIUM +EXPO

LUGER RESEARCH
Institute for Innovation & Technology

Context

- Digitization follows LEDification
  - LEDs allow/need control for efficient use
  - LED driver and controller integrated into luminaries
  - Control of luminaries over networks
- The promise
  - Many light sources
  - Expressive, structured, reactive, « smart » light
- Traditional control
  - A hierarchy of networks, cables ...
  - Centralized logic
- Distributed control
  - Push decisions to the luminaries themselves
  - Allow them to cooperate among each other
  - Creates commissioning free or easy platforms : plug and play
  - Scalable: no practical limit to network size (thousands to millions of independent nodes)
  - Robust: graceful degradation ...
- LED‘s CHAT : distributed, scalable, real time

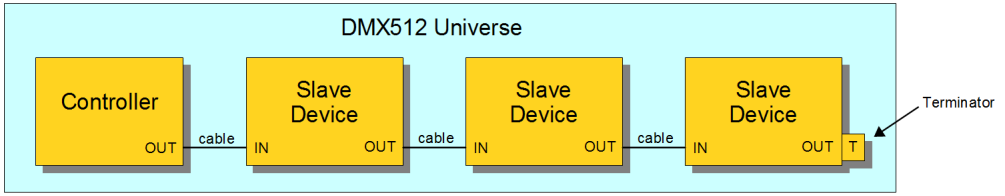
Historical decentralized control

- Light switch
- PIR sensor ...
  - Local decision and effect

©Dr.E. sur flickr

- No connections between luminaries

Classical centralized control

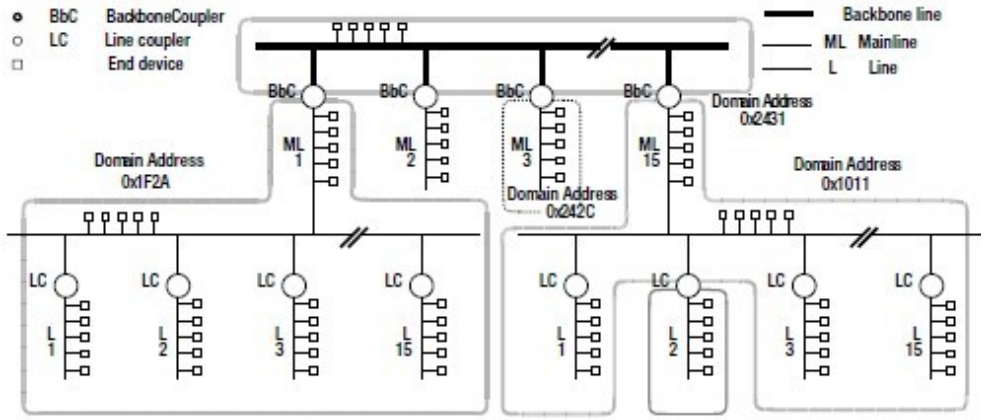


- Interconnection bus
  - (Multi-)Master – slave approach
  - Bus shared for all messages
  - Slaves interprete and execute commands addressed to them
  - Slaves may answer to the master

- Examples
  - DaLi (slow and robust)
  - DMX (not so slow, but very limited)
  - IP based systems (not so limited, but setup remains complicated)

Hierarchical networks

- Problem :
  - Limits of backend networks
- Examples :
  - DaLi (64) DMX (512) address spaces
  - Bus bandwidth



Source: Industrial Ethernet Book Issue 69 / 35

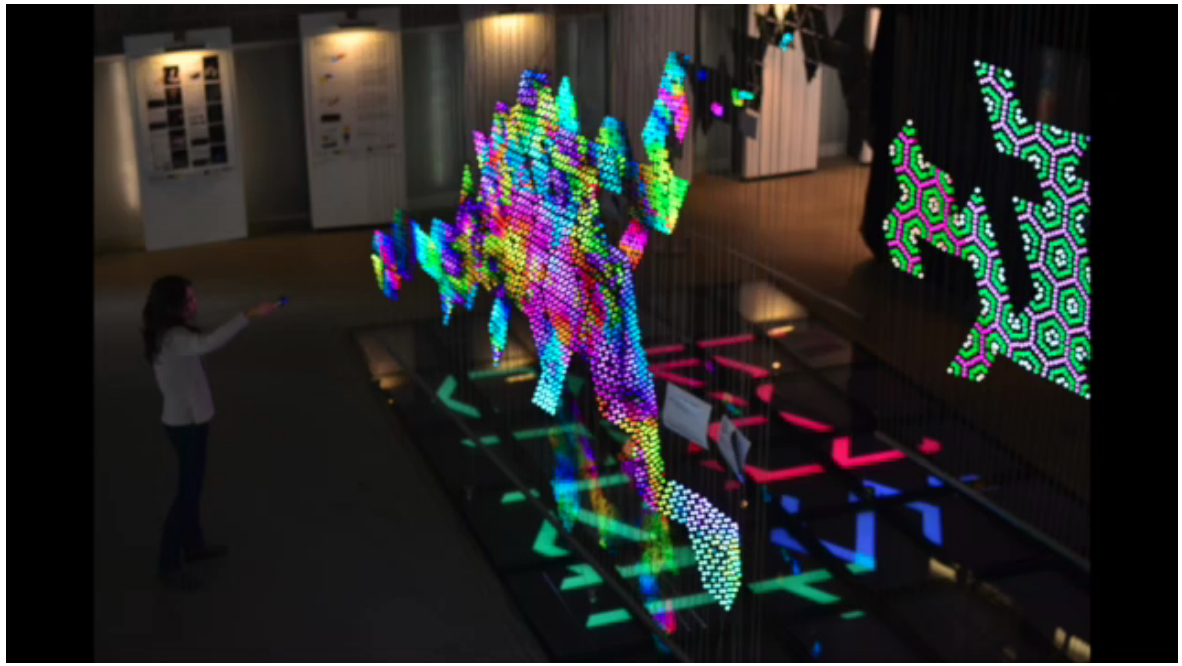
- Frequent solution :
  - « Groups » to execute the same command sent just once
  - Backbone networks and gateways
    - KNX tree topology, KNX to DaLi gateways
    - IP backbones
  - Separate solutions for commissioning (if not fully manual)
  - Zoo of expensive controllers

« Zoo » of controllers really needed?

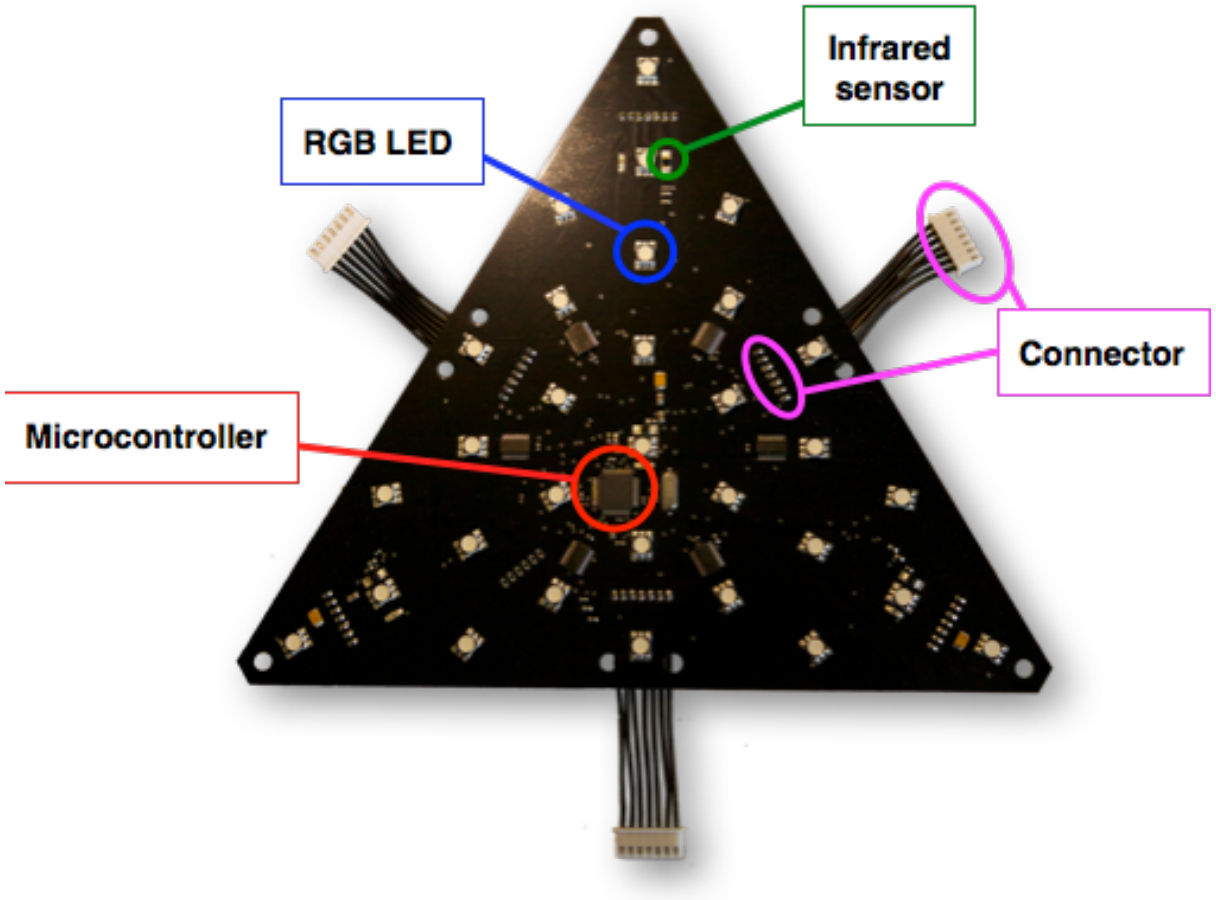
leds chat

- Luminaries could be smarter without additional cost
  - Luminaries are already equipped with an underused microcontroller and communication hardware
  - What if THEY form the backbone network?
  - What if THEY could communicate with each other?! **LED's CHAT!**
- Neighboring luminaries do related work
  - Topology can be meaningful.
- Origins of LED's CHAT
  - Started as a university project
  - Prototype installation at a cultural event (Marseille European Capital of Culture)
  - Academic freedom from standards : we could try something radically different.
  - Current philosophy : interoperable on the outside, proprietary (and free) inside

- Prototype created in 2013
  - 500 modules, each with 31 RGB LEDs and 4 IR sensors
  - Installation exposed during 3 months in Marseille, > 4000 visitors
  - Interaction with « torches »



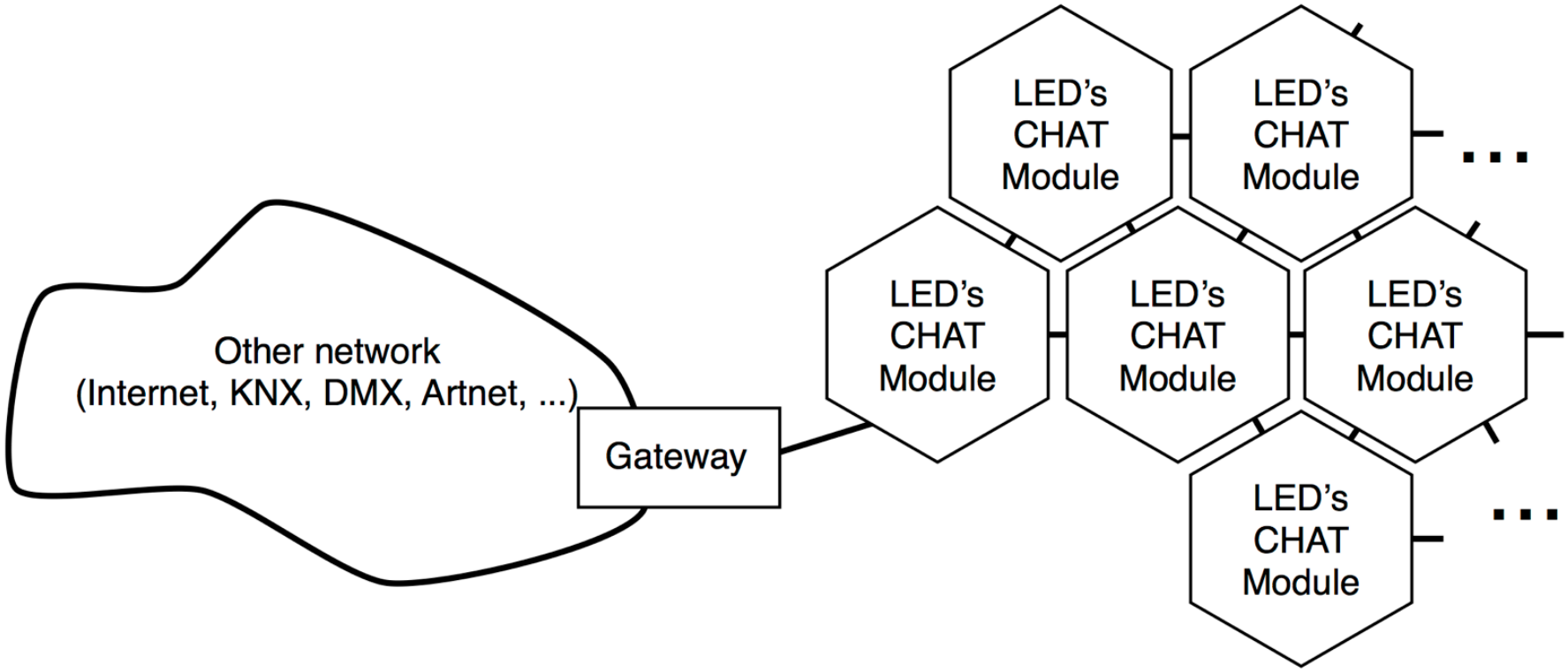
What's on the modules



In boxes and with diffusers …

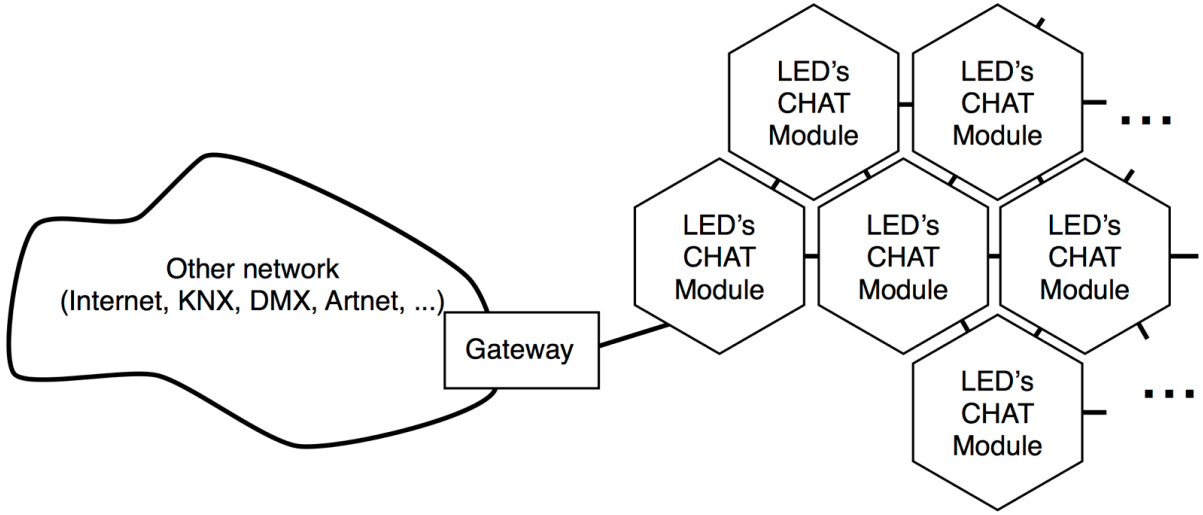
leds chat

LED‘s CHAT installation scenario



What is distributed control

- Centralized control
  - Master takes decisions for each module and emits commands
  - Network transports commands to modules
  - Modules/Slaves execute commands
- Distributed control
  - Modules execute control code and take decisions themselves
  - Network allows modules to communicate with each other and with Master
  - Master still emits global decisions and data, but much less than in centralized control



Centralized or distributed?

© woodleywonderworks, sur flickr

Advantages of distributed control

- Much reduced bandwidth between master and modules
  - Can be close to zero for fully distributed applications
- As a result, nearly full scalability :
  - As you add more modules
    - you get more computing power
    - you get more local bandwidth
  - Local communication means no interference, as opposed to bus architecture
- With less bandwidth, you get more!
  - The power of distributed algorithms!
  - Cooperation between neighbors!
- ChatOS (our distributed operating system)
  - Real time behavior with garanteed bandwidth for neighbors
  - Quasisynchronous operation of modules

Distributed algorithms?!

leds@chat

- What are they?!
  - Tasks performed together by the modules by computation and communication
- What can they do for you?!
  - Topology discovery and observation
  - Dynamic creation of « groups », e.g. variable zones in open spaces …
  - Synchronous execution with little communication
    - The conductor won't read the partition to the musician, both read it at the same time
  - Sensor fusion and local diffusion
  - Firmware upgrade, ephemeral application distribution
  - Classical tasks like broadcast and convergecast
    - Video diffusion, data collection …

Sensor fusion

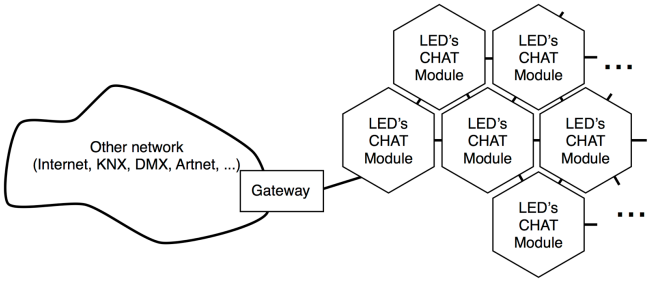
- Neighboring modules can share their sensor observations
  - Sensor fusion without the need of passing by the master/gateway
  - E.g. distance or presence sensors
  - Fusion for robustness and resolution, boundary detection, etc.
- Fully scalable, very low latency

- Speaking of sensors …
  - You already have a microcontroller with your luminary!
  - Why do you want another one for creating a sensor?!
  - Adding useful sensors to lighting modules is not expensive
- Sensors :
  - Ambient light, presence, movement … for applications
  - Temperature, voltage, current … for monitoring and failure prevention

Demonstration of Sensor Fusion

So what is the role of the gateway?

- Central view has many uses
  - Time base, route optimizations …, gateway has more computing power …
- Interoperability
  - via IP and software
  - via physical interfaces (KNX, DMX, …) where required
- Examples
  - KNX or DMX messages can be received and broadcast in the LED's CHAT network
    - LED's CHAT distributed application code specifies how to interprete
    - address or group membership can be statically or dynamically assigned by a distributed algorithm to each LED!
  - External information sources (video …) can be broadcasted into the network
  - Sensor data can be aggregated and recovered.



LED‘s CHAT Ecosystem

leds chat

- ChatOS Base
  - Lightweight distributed operating system for microcontrollers
  - Basic distributed system services
    - discovery, distributed synchronization (hot pluggable)
    - Network wide synchronous communication and execution
    - API for hardware access (sensor reading, LED control)
  - Basic gateway services
    - Application distribution and firmware update
    - topology discovery and tracking (hotplug / failures ...)
    - Dispatching of synchronous commands
    - Broadcast and convergecast services
- LED‘s CHAT Application model
  - Distributed synchronous programming model
  - Applications in neighboring modules advance synchronously
    - No need for handshake on application level!
    - Guaranteed synchronous communication between neighbors!

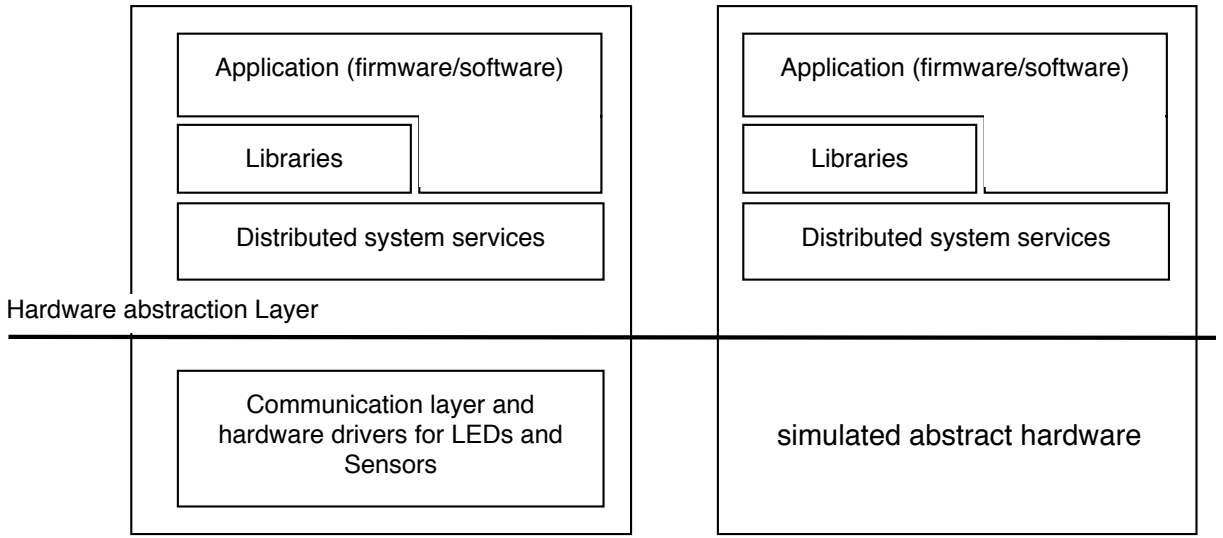
LED‘s CHAT Ecosystem simulation

leds chat

- ChatOS Base
  - Lightweight distributed operating system for microcontrollers
  - Basic distributed system services
    - discovery, distributed synchronization (hot pluggable)
    - Network wide synchronous communication and execution
    - API for hardware access (sensor reading, LED control)
  - Basic gateway services
    - Application distribution and firmware update
    - topology discovery and tracking (hotplug / failures ...)
    - Dispatching of synchronous commands
    - Broadcast and convergecast services
- LED‘s CHAT Application model
  - Distributed synchronous programming model (C++)
  - Applications in neighboring modules advance synchronously
    - No need for handshake on application level!
    - Guaranteed synchronous communication between neighbors!

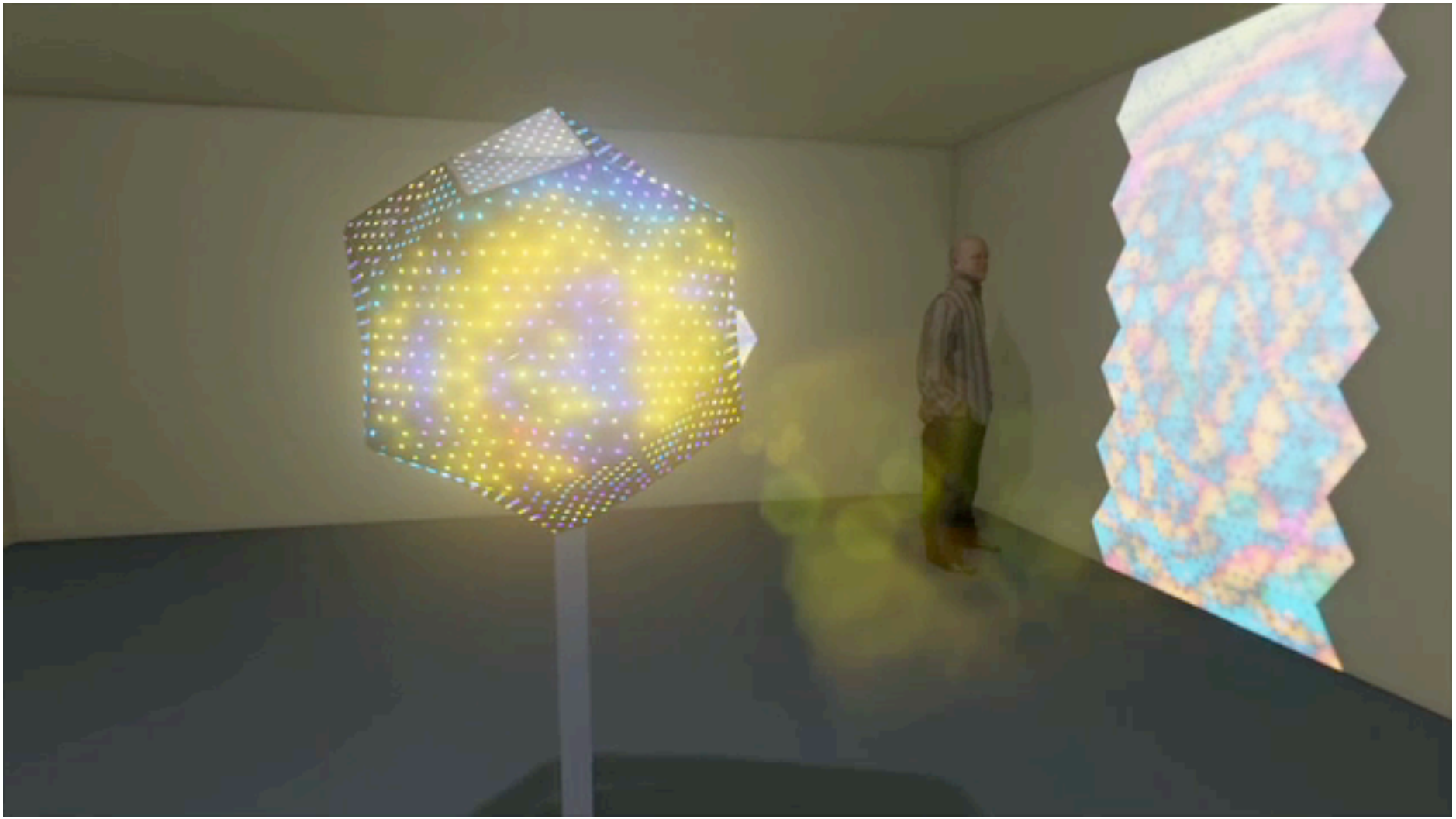
LED's CHAT Simulator

- Both application and operating system can be simulated
  - Except for low level synchronization and communication
- Real time simulation in Game Engine
- Realistic rendering (not real time) available
- Debugging of application code with VisualStudio



LED's CHAT Simulator

leds chat



Hardware requirements

- Mid range and lower end microcontrollers
  - Depends on the number of LEDs and the complexity of the control task
    - For one point sources Cortex M0/M0+ or equivalent is clearly sufficient
    - For multi pixel modules, midrange µC are a better choice (ARM Cortex M3, PIC32, …)

- That said …
  - an operational prototype with low end 8 bit micro controllers and 6 LEDs was our first experiment.

Some use cases

leds chat

- Ephemeral installations
  - Easy commissioning makes LED's CHAT great for exhibition spaces, etc
- Expressive interactive lighting
  - Interaction is immediate and low latency, great user experience
  - High quality transitions due to distributed effect generation
- Open space ceiling light
  - Dynamic zones, ambient lighting for well being …
- Total cost of ownership advantages
  - Modularity, easy commissioning (plug and play)
  - Graceful degratation in case of component or connection failure
  - Component failure can be automatically detected and signaled

LED‘s CHAT, the company

- 2015 startup
  - The CEO (business development), two software engineers, a scientific advisor (me)
- Current status :
  - Consolidating technology towards commercial level
  - Discussing with potential customers and partners
- Looking for partnerships of several kinds
  - Pilot projects
  - Industrialization, co-development
  - Investment
- Interested?

Conclusions

- Distributed control is possible without additional hardware
  - But respecting current standards, potential is rather limited
  - LED's CHAT is proprietary, but interoperable (gateway)
- Distributed control has many advantages
  - Easy commissioning, graceful degradation
  - More efficient use of bandwidth
  - Scalability
- Adoption
  - To some extend, partial adoption in many products (auto addressing)
  - LED's CHAT goes further and distributes the essential part of control software
- LED's CHAT system
  - Proof of concept prototype
  - Maturing software ecosystem
  - Simulator for development and project support

leds chat

Thank you!