

**Stijn Denis – Ben Bellekens**



IDLab  
INTERNET & DATA LAB

Universiteit  
Antwerpen

umec istart

umec  
embracing a better life

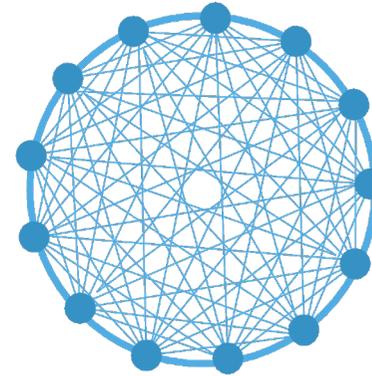


**CROWDSCAN**  
THE FUTURE OF CROWD ANALYTICS



# OUR SOLUTION

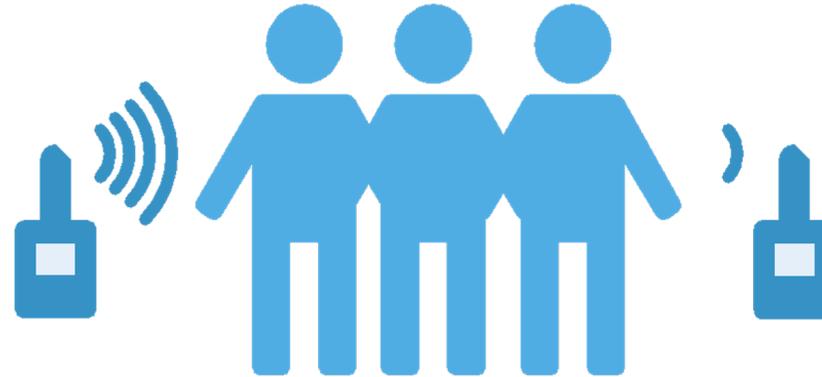
**CROWDSCAN** measures the average attenuation of a wireless sensor network relative to the empty environment.



RF Transmission area with sensors



RF Transmission through body

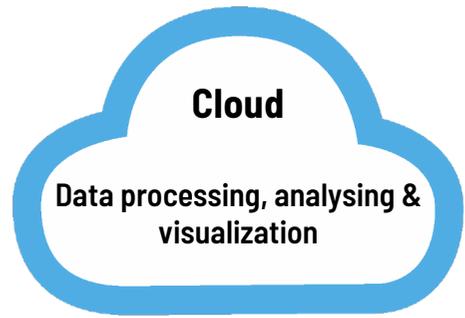
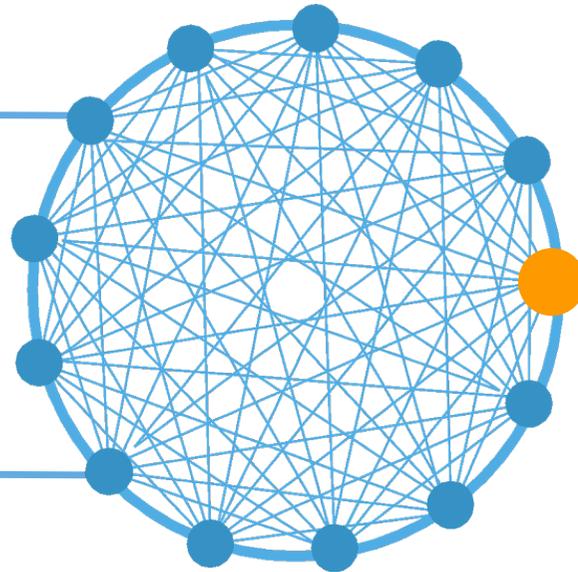


# OUR TECHNOLOGY HARDWARE

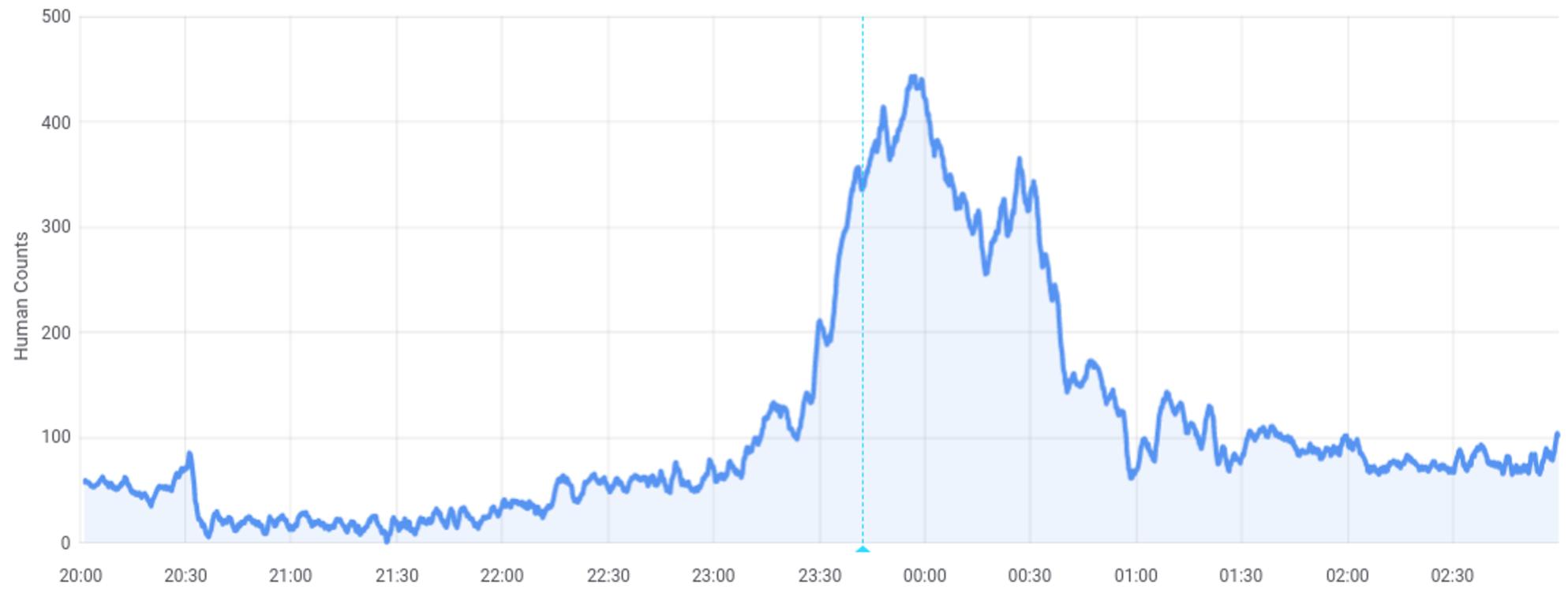
**Node**  
Waterproof,  
Battery-  
powered

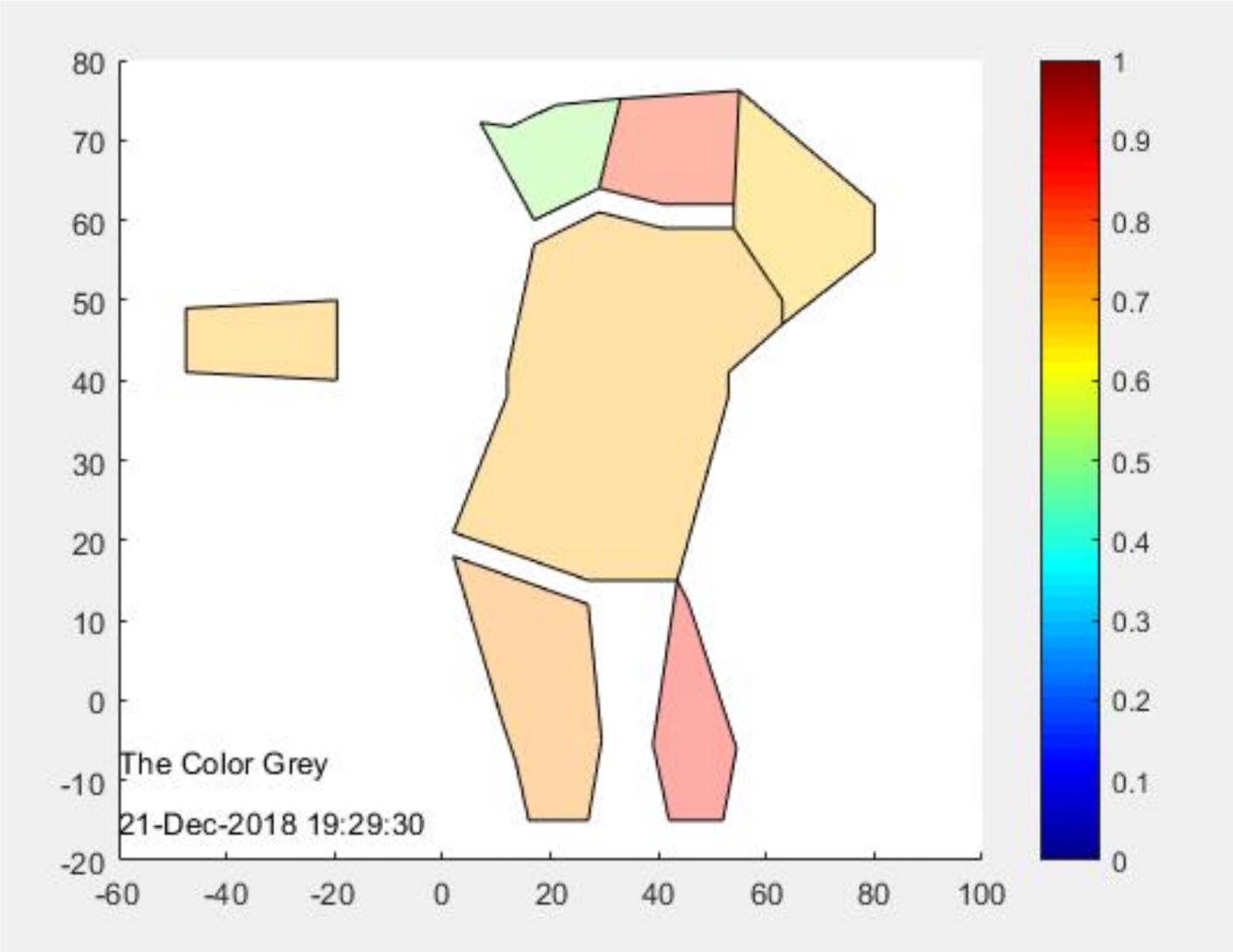


**Node**  
Waterproof,  
Battery-  
powered

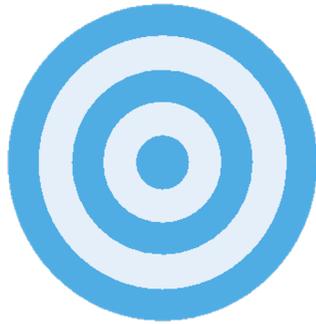


**Gateway**  
Powered Network





# OUR KEY MARKET DIFFERENTIATORS



**Accurate**



**Real-time**

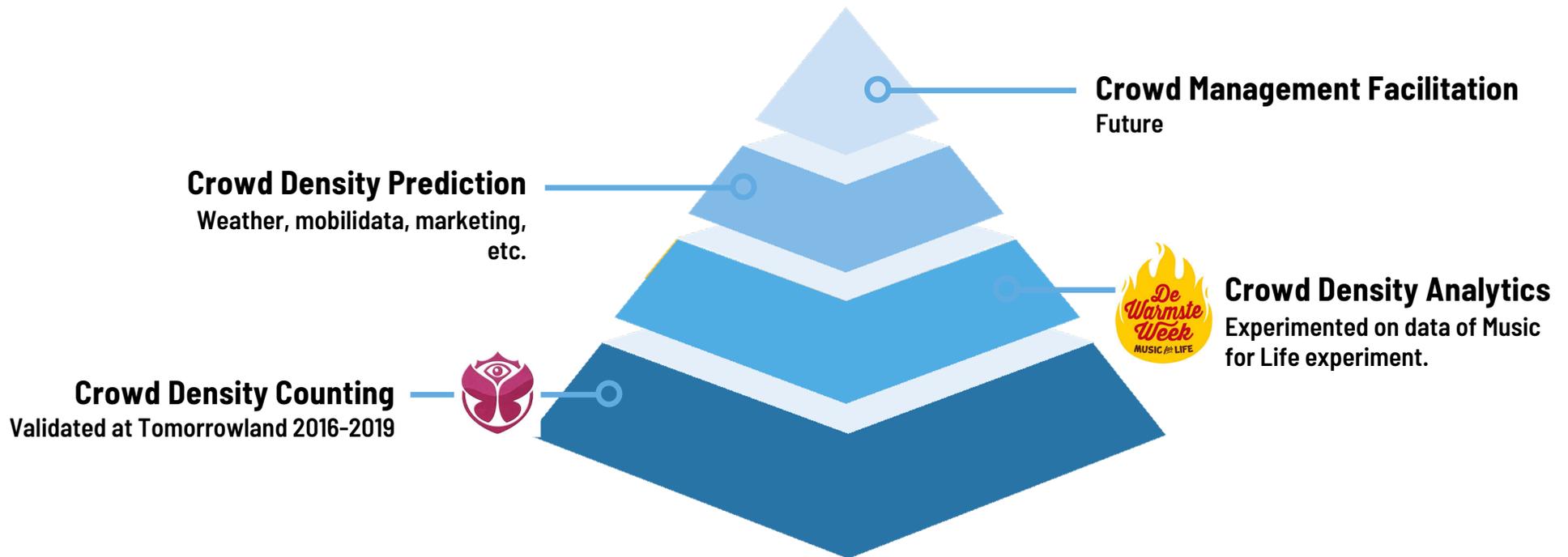


**Privacy-conscious**



**Tagless/Device  
-Free**

# OUR ROADMAP



# Behind the scenes

**How do we estimate crowd density in real-time ?**

**How accurate is this ?**

**Where and when can we expect critical situations?**

**Where are anomalies ?**

**Which action should we take ?**

**How can we solve this for next time and can we measure it in the same way so that we get alerted?**





## Crowd Analytics

When and where does the crowd move to?

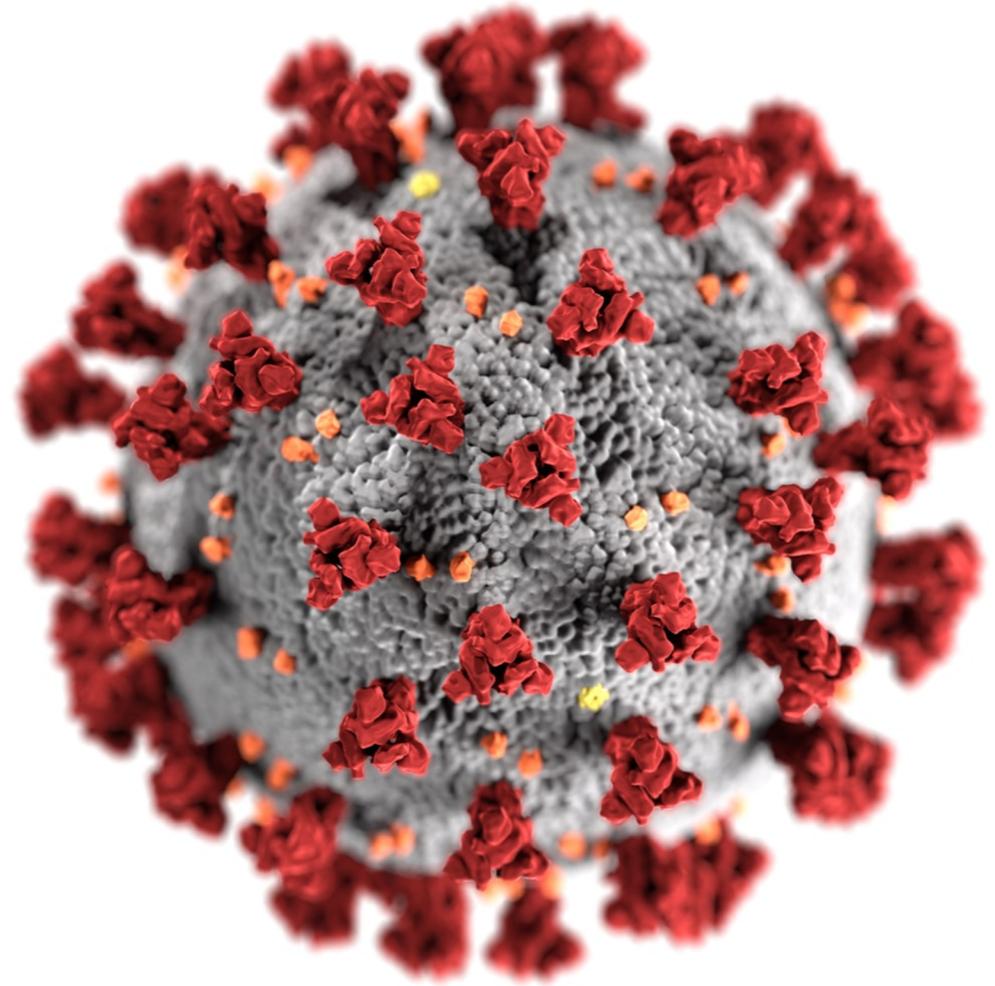
Which flow will probably change in the future when action x happens?

How many visitors are there more than a regular day?

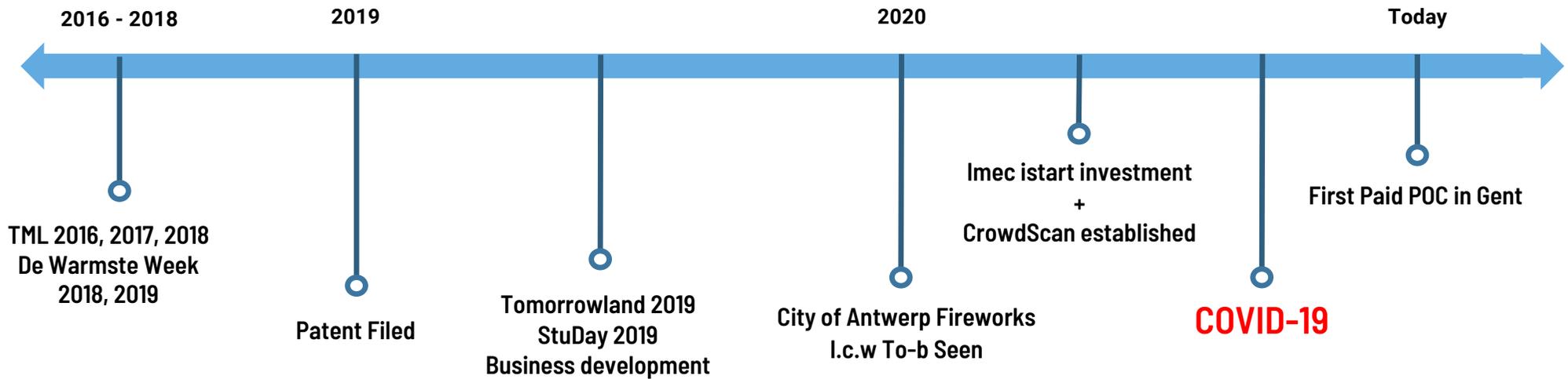
Which store is the most visited one?

# COVID-19 Virus

- Measures
  - a. Social Distancing (1.5m)
    - i. 0.27 personen/m<sup>2</sup>
    - ii. 3.6 m<sup>2</sup> per persoon
- Monitoring
  - a. Capacity
  - b. Density barometer

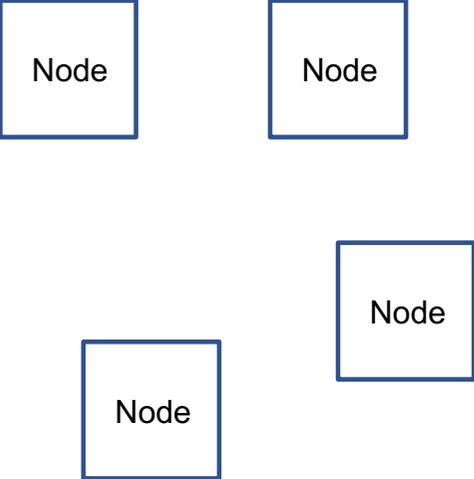


# OUR PROJECT STATUS



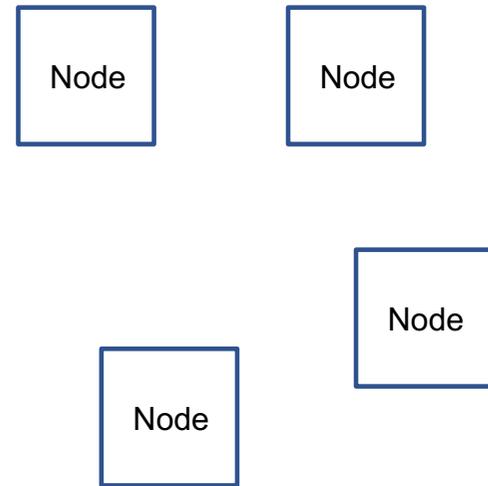
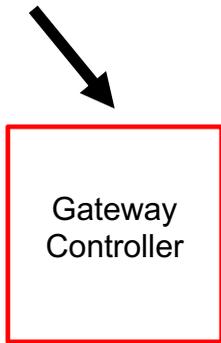
# **TAGLESS CROWD ESTIMATION USING DASH7**

# CONFIGURATION



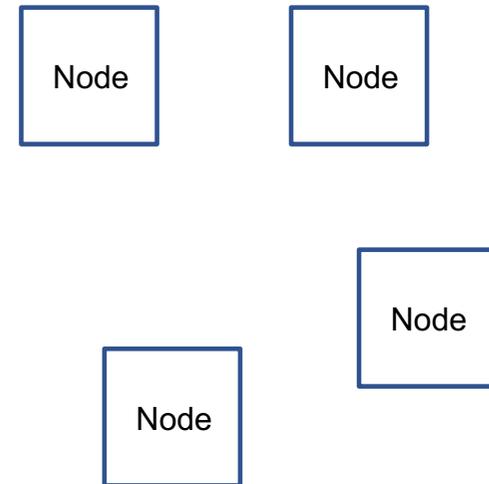
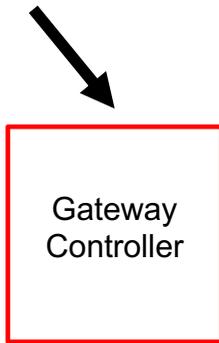
# CONFIGURATION

1) Write configuration parameters to files



# CONFIGURATION

1) Write configuration parameters to files

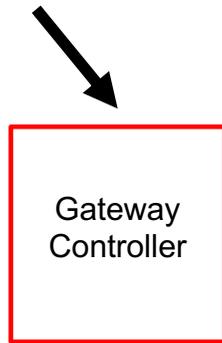


## Configuration Parameters:

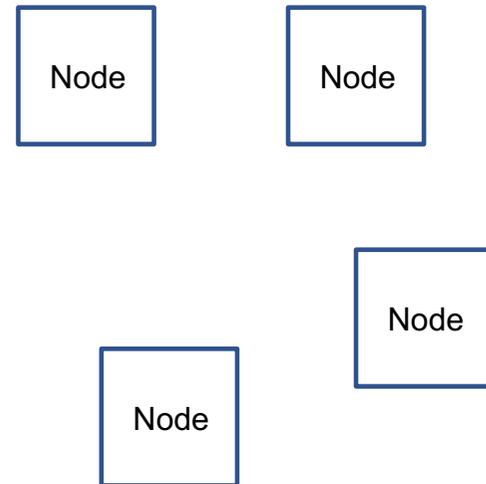
- 1) 2 Access Classes for use in Phase 2 (Foreground Scanning & Background Scanning)
- 2) Application-specific parameters (wtime, network\_size)
- 3) UID – VID list

# CONFIGURATION

1) Write configuration parameters to files



2) Nodes will perform file read requests to obtain all configuration parameters

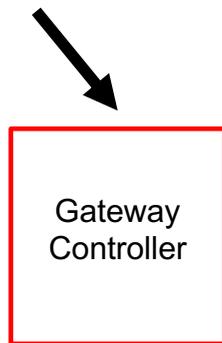


## Configuration Parameters:

- 1) 2 Access Classes for use in Phase 2 (Foreground Scanning & Background Scanning)
- 2) Application-specific parameters (wtime, network\_size)
- 3) UID – VID list

# CONFIGURATION

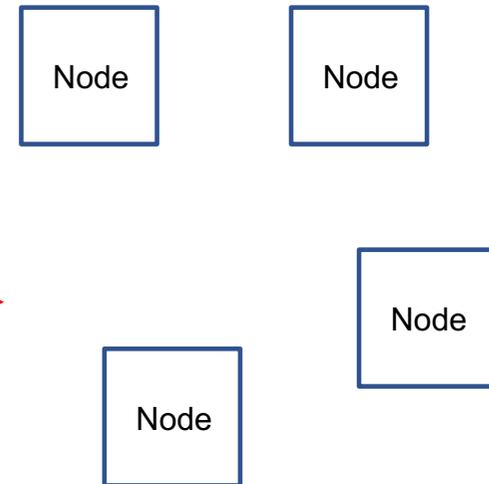
1) Write configuration parameters to files



2) Nodes will perform file read requests to obtain all configuration parameters



3) Controller responds to these requests

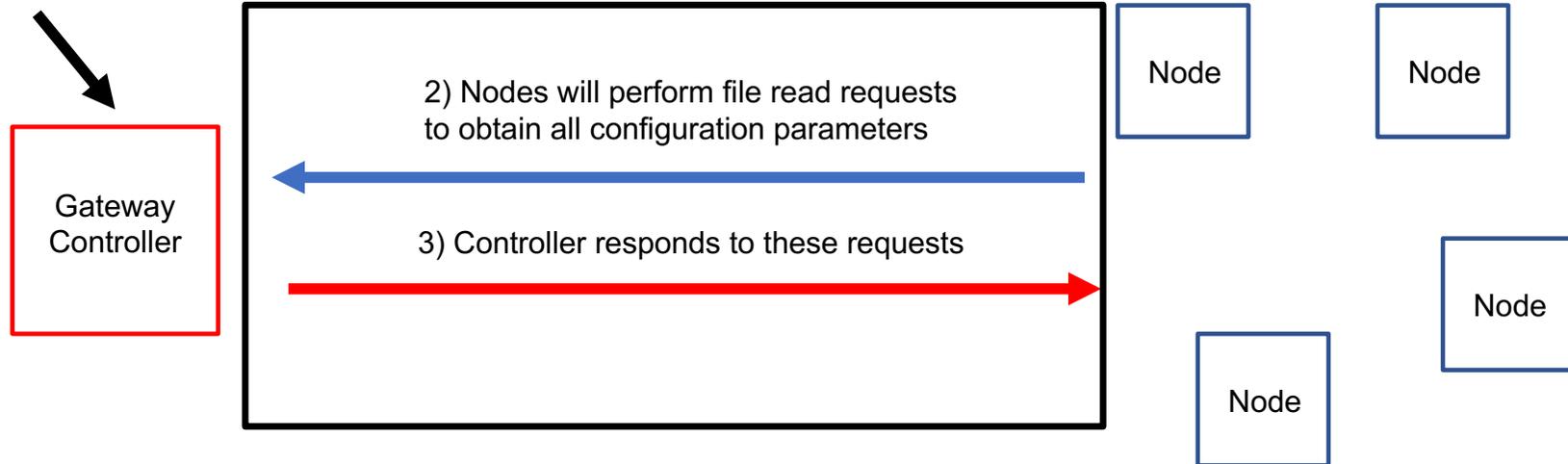


## Configuration Parameters:

- 1) 2 Access Classes for use in Phase 2 (Foreground Scanning & Background Scanning)
- 2) Application-specific parameters (wtime, network\_size)
- 3) UID – VID list

# CONFIGURATION

1) Write configuration parameters to files



## Configuration Parameters:

- 1) 2 Access Classes for use in Phase 2 (Foreground Scanning & Background Scanning)
- 2) Application-specific parameters (wtime, network\_size)
- 3) UID – VID list

# FILE READ REQUEST

All communication during the configuration phase occurs through the use of a pre-defined AC known by all nodes and the controller.

The ACs meant to be used during actual operation of the system & the app-specific parameters can be transmitted as a regular response to the file read requests.

Each node also requests a unique VID. The controller has a user file which contains a table linking each UID to a VID. A node will transmit a read request for a nonexistent file. This request will be sent upwards to the application layer where it can be handled in such a manner as to only respond with the correct VID for the requesting node.

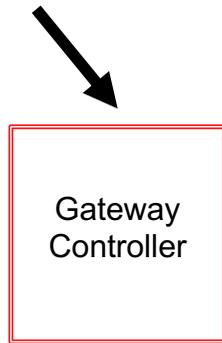


|   | OSI Layer | D7A Component  | Description   |
|---|-----------|--|---|
| ALP   | 7         | Application  | File Access<br>File management via Application Layer Programming Interface (ALP). Property-based queries.<br><br>D7AActP<br>Pre-registered application actions, triggered conditionally on file access. |
|   | 6         | Data Elements  | D7A Files<br>A user-driven file system that supports read, write, create, delete, modify of executable, encryptable files.<br><br>Native attributes allowing to configure D7AActP                       |
| D7A   | 5         | Session  | D7ASP<br>Session FIFOs, priorities and QoS management.  |
|   | 4         | Transport  | D7ATP<br>Request-Response and group acknowledgment.   |
|   | 3         | Network  | D7ANP<br>Routing (no-hop & one-hop), foreground scan automation   |
|   |           |  | D7AAdvP<br>Ad-hoc synchronization   |
|   |           |  | Addressing<br>Access Profile  |
| Security<br>AES-128 authentication and encryption |           |  |   |
| 2   | Data Link | Frame Addressing<br>Unicast, Broadcast<br><br>Transmission<br>Upper-layer event driven<br><br>Data Reception<br>Upper-layer event driven, or via configurable, sequential automated channel scan<br><br>Channel Access<br>CSMA-CA, with static channel guarding rules, multiple supported flow/congestion control models and frequency diversity.                                    |   |
| 1   | Physical  | Channel QoS<br>Clear Channel Assessment<br><br>Encoding<br>1/1 PN9, 1/2 convolutional Code<br><br>Rates<br>9.6 kb/s, 55.55 kb/s, 166.667 kb/s<br><br>Modulation<br>$\pm 4.8$ kHz 2-(G)FSK, $\pm 50$ kHz 2-(G)FSK or $\pm 41.667$ kHz 2-(G)FSK<br><br>Channel spacing<br>25 kHz or 200 kHz<br><br>Spectrum<br>433.060 – 434.785 MHz<br>863.000 – 870.000 MHz<br>902.000 – 928.000 MHz |   |

Table 4.1.1: DASH7 Alliance Protocols Stack Overview

# CONFIGURATION

1) Write configuration parameters to files



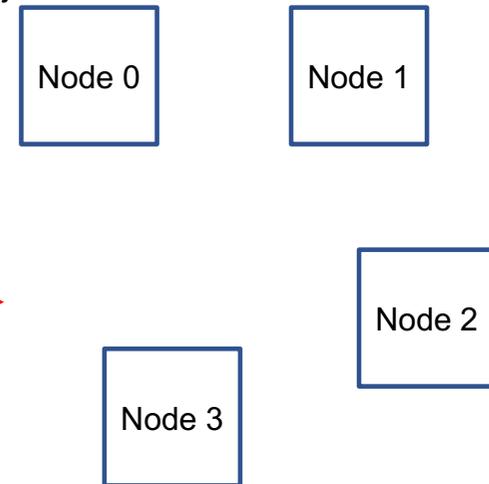
2) Nodes will perform file read requests to obtain all configuration parameters



3) Controller responds to these requests



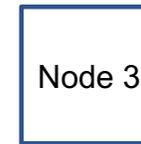
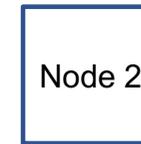
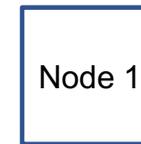
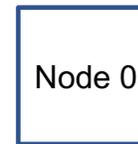
4) Once the nodes have received their configuration, they will switch to the background scanning access class they just received



## Configuration Parameters:

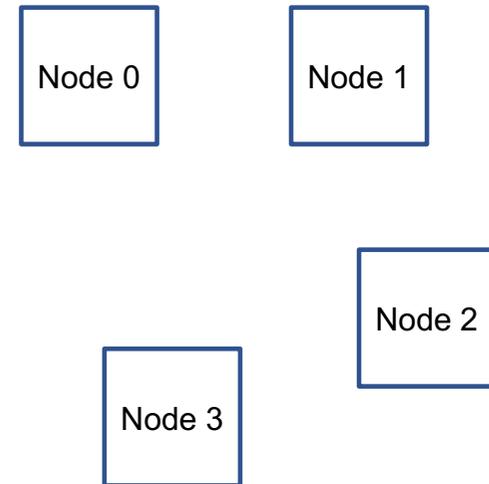
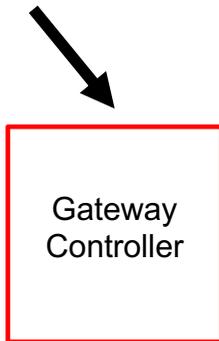
- 1) 2 Access Classes for use in Phase 2 (Foreground Scanning & Background Scanning)
- 2) Application-specific parameters (wtime, network\_size)
- 3) UID – VID list

# REGULAR OPERATION (PHASE 2) - START



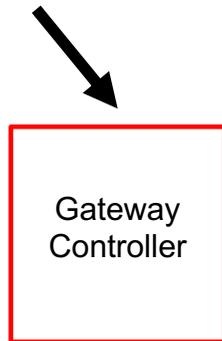
# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller

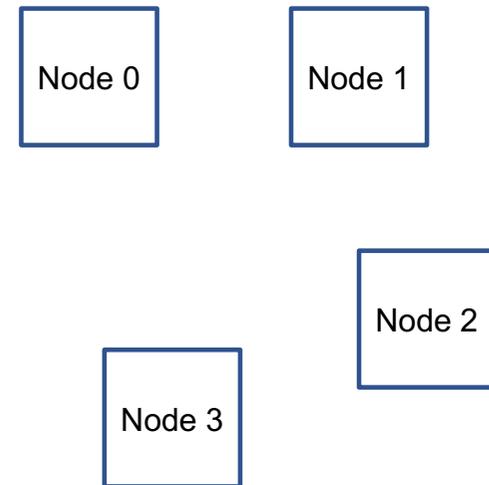


# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller



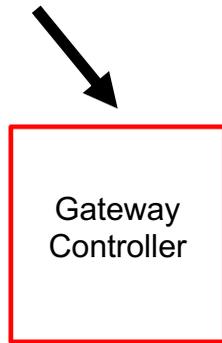
2) A 'START' command is sent to the Controller



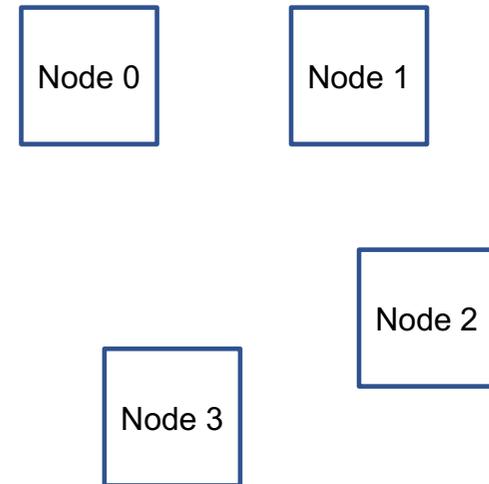
**All commands amount to writing a (set of) value(s) to a specific user file, which triggers an action in the application layer.**

# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller



3) The corresponding AC is used to transmit BG frames and perform a low-power wakeup of the BG-scanning nodes

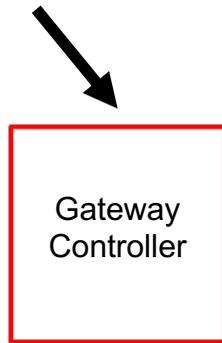


2) A 'START' command is sent to the Controller

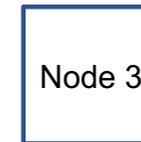
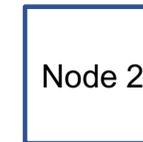
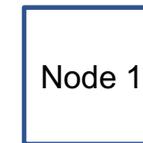
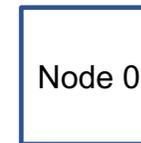
**All commands amount to writing a (set of) value(s) to a specific user file, which triggers an action in the application layer.**

# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller



3) The corresponding AC is used to transmit BG frames and perform a low-power wakeup of the BG-scanning nodes  
4) A predefined 'START' message is broadcasted to the nodes



2) A 'START' command is sent to the Controller



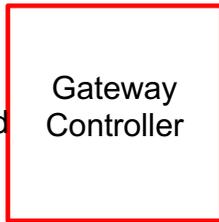
**All commands amount to writing a (set of) value(s) to a specific user file, which triggers an action in the application layer.**

# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller

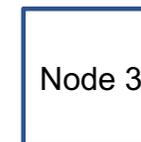
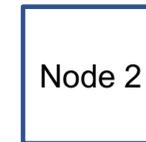
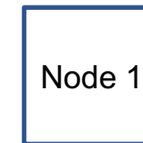
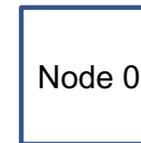


5) The Controller switches its active AC to a foreground scan



2) A 'START' command is sent to the Controller

3) The corresponding AC is used to transmit BG frames and perform a low-power wakeup of the BG-scanning nodes  
4) A predefined 'START' message is broadcasted to the nodes



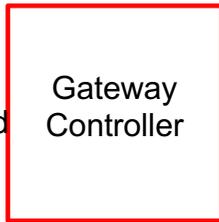
**All commands amount to writing a (set of) value(s) to a specific user file, which triggers an action in the application layer.**

# REGULAR OPERATION (PHASE 2) - START

1) An 'ENABLE\_PHASE\_2' command is sent to the Controller



5) The Controller switches its active AC to a foreground scan

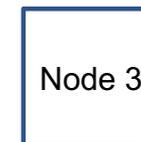
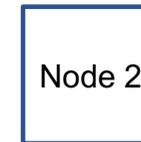
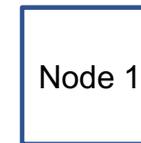
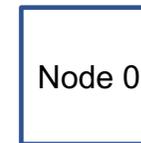


2) A 'START' command is sent to the Controller

3) The corresponding AC is used to transmit BG frames and perform a low-power wakeup of the BG-scanning nodes  
4) A predefined 'START' message is broadcasted to the nodes

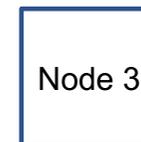
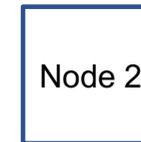
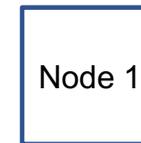
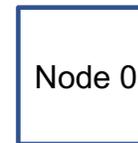


6) After receiving the 'START' message, The nodes switch their active AC to a foreground scan



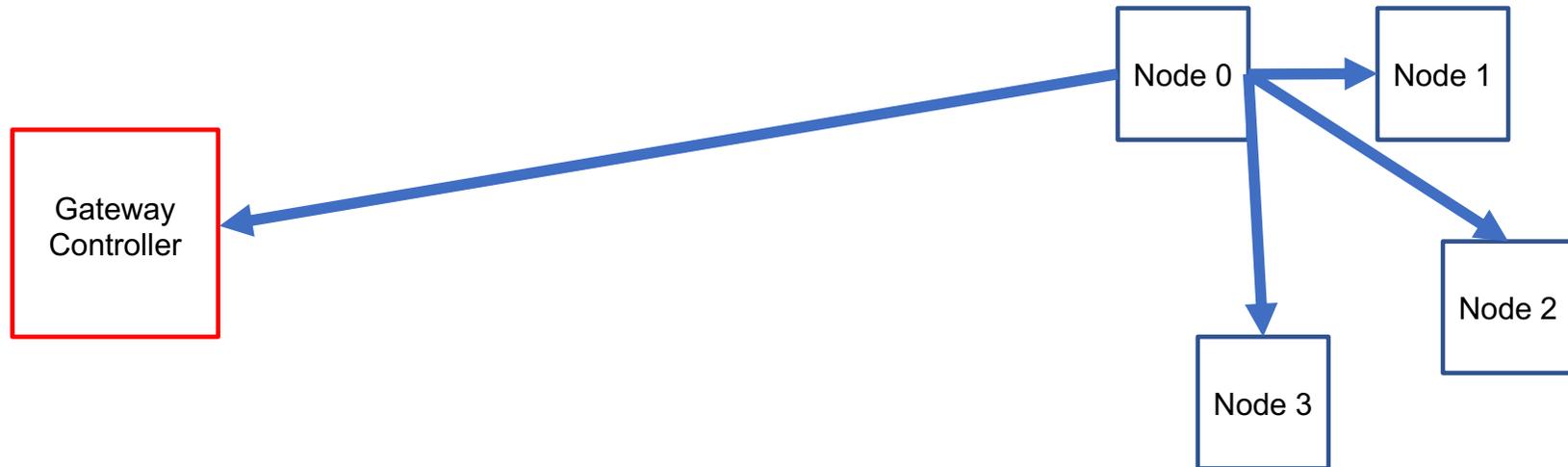
**All commands amount to writing a (set of) value(s) to a specific user file, which triggers an action in the application layer.**

# REGULAR OPERATION (PHASE 2) - CYCLE



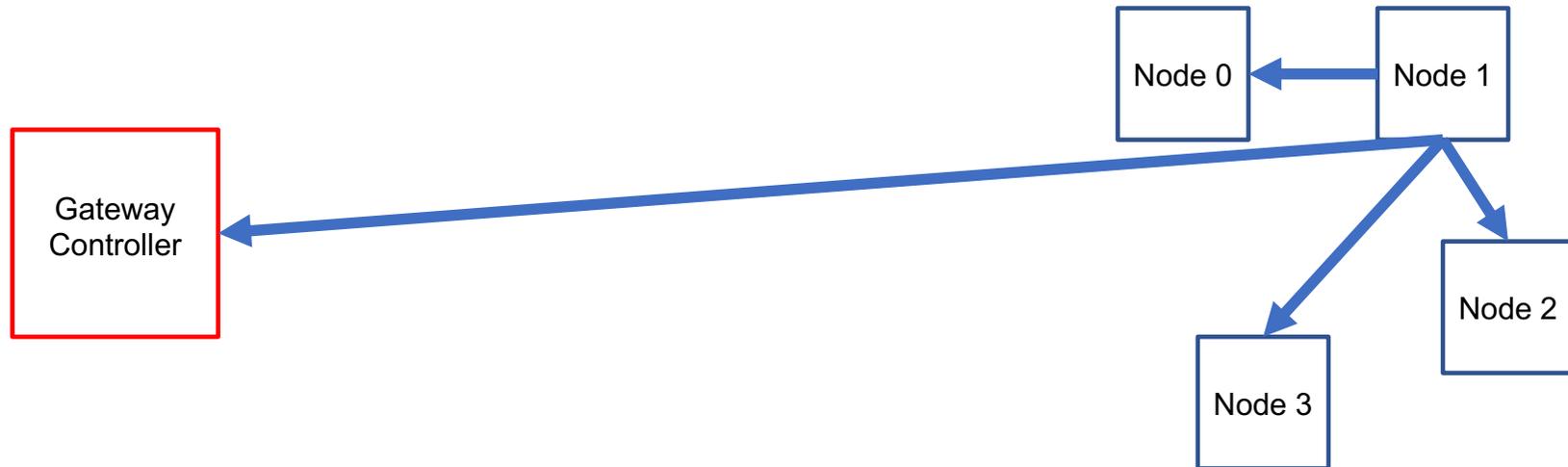
# REGULAR OPERATION (PHASE 2) - CYCLE

1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



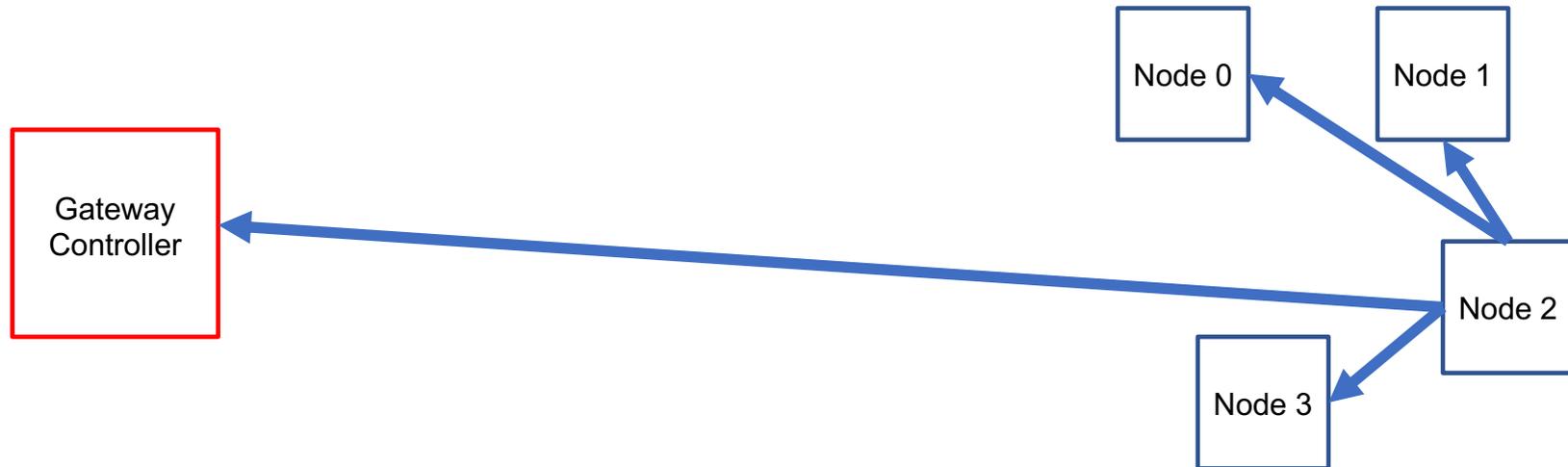
# REGULAR OPERATION (PHASE 2) - CYCLE

1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



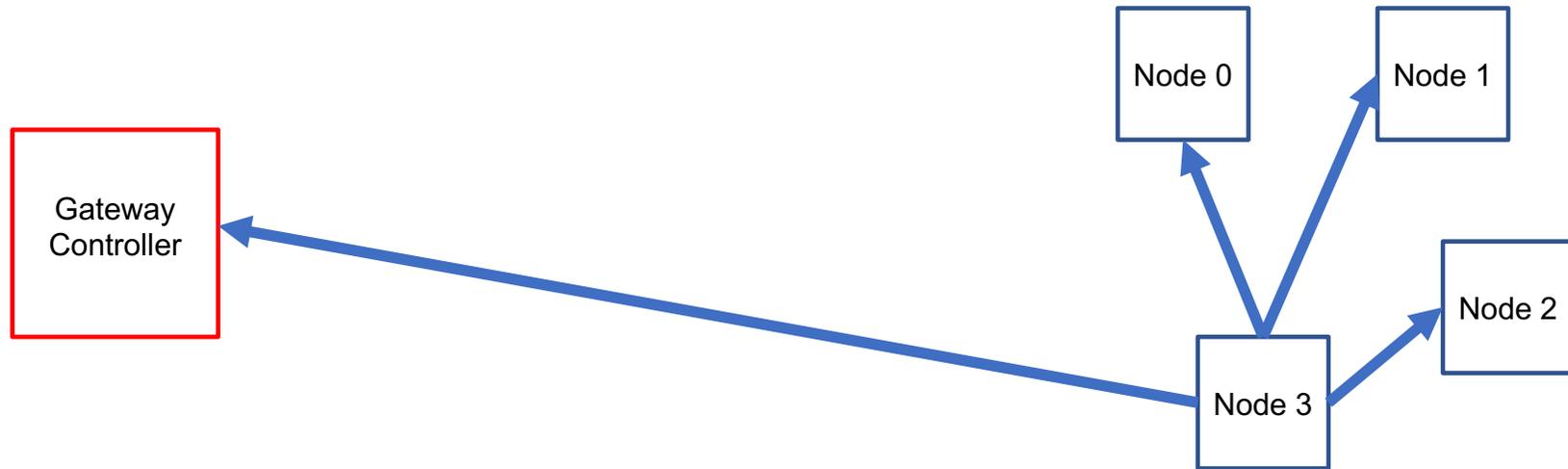
# REGULAR OPERATION (PHASE 2) - CYCLE

1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



# REGULAR OPERATION (PHASE 2) - CYCLE

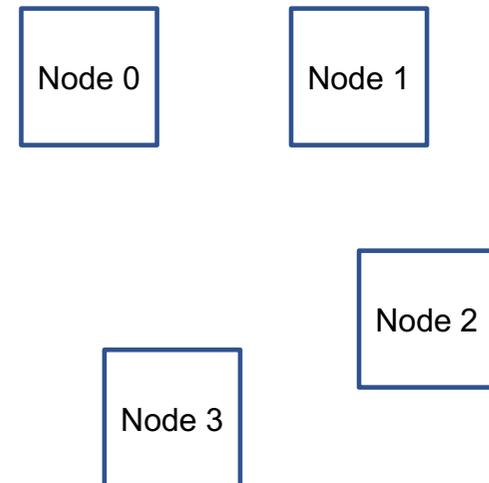
1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



# REGULAR OPERATION (PHASE 2) - CYCLE



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller

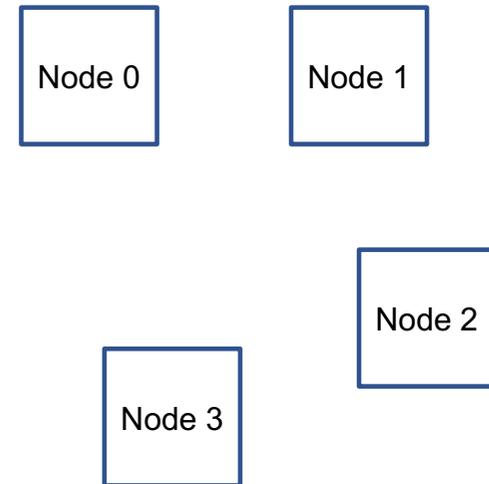


2) The RSS with which these messages are received, are stored in an internal list for each node

# REGULAR OPERATION (PHASE 2) - CYCLE



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



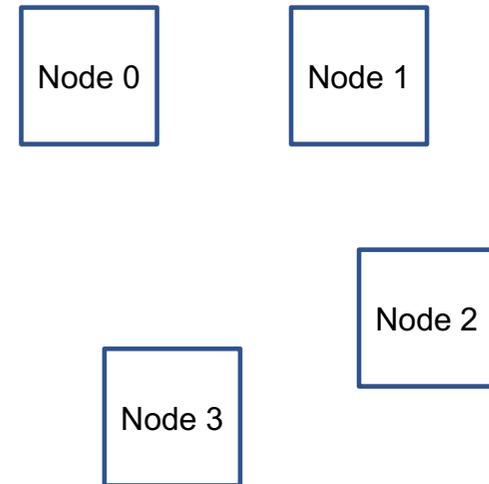
2) The RSS with which these messages are received, are stored in an internal list for each node

3) The payload of these messages consists of this internal list

# REGULAR OPERATION (PHASE 2) - CYCLE



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



2) The RSS with which these messages are received, are stored in an internal list for each node

3) The payload of these messages consists of this internal list

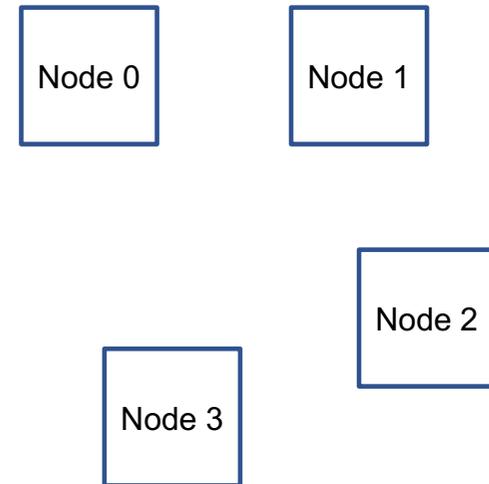
4) After broadcasting, all list values are reset to 0

# REGULAR OPERATION (PHASE 2) - CYCLE

5) The controller receives all messages sent by the nodes and therefore obtains all RSS measurements



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



2) The RSS with which these messages are received, are stored in an internal list for each node

3) The payload of these messages consists of this internal list

4) After broadcasting, all list values are reset to 0

# REGULAR OPERATION (PHASE 2) - CYCLE

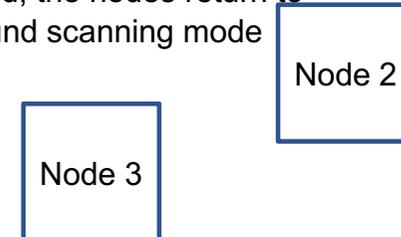
5) The controller receives all messages sent by the nodes and therefore obtains all RSS measurements



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



6) Once a full cycle has been completed, the nodes return to background scanning mode



2) The RSS with which these messages are received, are stored in an internal list for each node

3) The payload of these messages consists of this internal list

4) After broadcasting, all list values are reset to 0

# REGULAR OPERATION (PHASE 2) - CYCLE

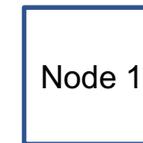
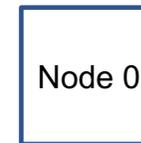
5) The controller receives all messages sent by the nodes and therefore obtains all RSS measurements



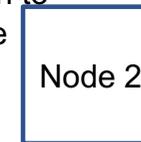
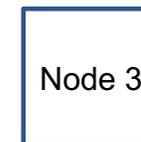
7) The Controller performs a low-power wakeup and the cycle begins anew



1) Each in turn, the nodes broadcast a message which can be received by all other nodes and the Controller



6) Once a full cycle has been completed, the nodes return to background scanning mode



2) The RSS with which these messages are received, are stored in an internal list for each node

3) The payload of these messages consists of this internal list

4) After broadcasting, all list values are reset to 0

# PHASE 2 MESSAGES

All ALP messages during phase 2 are responses to nonexistent requests (0x20).

For the nodes, these messages are sent upwards to the application layer where they can be dealt with accordingly.

For the Controller, these messages are sent to their UART interface. In doing so, all RSS measurement data is sent to the computing unit of the gateway to which the Controller module is connected.



|   | OSI Layer | D7A Component  | Description   |
|---|-----------|--|---|
| ALP   | 7         | Application  | File Access<br>File management via Application Layer Programming Interface (ALP). Property-based queries.<br>D7AActP<br>Pre-registered application actions, triggered conditionally on file access. |
|   | 6         | Data Elements  | D7A Files<br>A user-driven file system that supports read, write, create, delete, modify of executable, encryptable files.<br>Native attributes allowing to configure D7AActP                       |
| D7A   | 5         | Session  | D7ASP<br>Session FIFOs, priorities and QoS management.  |
|   | 4         | Transport  | D7ATP<br>Request-Response and group acknowledgment.   |
|   | 3         | Network  | D7ANP<br>Routing (no-hop & one-hop), foreground scan automation   |
|   |           |  | D7AAdvP<br>Ad-hoc synchronization   |
|   |           |  | Addressing<br>Access Profile  |
| Security<br>AES-128 authentication and encryption |           |  |   |
| 2   | Data Link | Frame Addressing<br>Unicast, Broadcast<br>Transmission<br>Upper-layer event driven<br>Data Reception<br>Upper-layer event driven, or via configurable, sequential automated channel scan<br>Channel Access<br>CSMA-CA, with static channel guarding rules, multiple supported flow/congestion control models and frequency diversity.                            |   |
| 1   | Physical  | Channel QoS<br>Clear Channel Assessment<br>Encoding<br>1/1 PN9, 1/2 convolutional Code<br>Rates<br>9.6 kb/s, 55.55 kb/s, 166.667 kb/s<br>Modulation<br>$\pm 4.8$ kHz 2-(G)FSK, $\pm 50$ kHz 2-(G)FSK or $\pm 41.667$ kHz 2-(G)FSK<br>Channel spacing<br>25 kHz or 200 kHz<br>Spectrum<br>433.060 – 434.785 MHz<br>863.000 – 870.000 MHz<br>902.000 – 928.000 MHz |   |

Table 4.1.1: DASH7 Alliance Protocols Stack Overview

# PHASE 2 COMMANDS

In addition to 'START', several other commands exist to manage the network.

'STOP' causes the Controller to cease performing low-power wakeups and sending 'START' messages after the end of the current cycle.

'DISABLE\_PHASE\_2' causes the controller to revert back to configuration mode.

'FULL\_RECONFIGURE' causes the controller to perform a low-power wakeup and transmit a specific 'CONFIGURATION\_RESET' message to the nodes, which makes them revert back to their initial configuration mode. This is useful for changing the ACs after installation.

| OSI Layer |   | D7A Component | Description  |
|-----------|---|---------------|--|
| ALP       | 7 | Application   | File Access<br>File management via Application Layer Programming Interface (ALP). Property-based queries.<br><br>D7AActP<br>Pre-registered application actions, triggered conditionally on file access.  |
|           | 6 | Data Elements | D7A Files<br>A user-driven file system that supports read, write, create, delete, modify of executable, encryptable files.<br><br>Native attributes allowing to configure D7AActP  |
| D7A       | 5 | Session       | D7ASP<br>Session FIFOs, priorities and QoS management.   |
|           | 4 | Transport     | D7ATP<br>Request-Response and group acknowledgment.  |
|           | 3 | Network       | D7ANP<br>Routing (no-hop & one-hop), foreground scan automation<br><br>D7AAdvP<br>Ad-hoc synchronization<br><br>Addressing<br>Access Profile<br><br>Security<br>AES-128 authentication and encryption  |
|           | 2 | Data Link     | Frame Addressing<br>Unicast, Broadcast<br><br>Transmission<br>Upper-layer event driven<br><br>Data Reception<br>Upper-layer event driven, or via configurable, sequential automated channel scan<br><br>Channel Access<br>CSMA-CA, with static channel guarding rules, multiple supported flow/congestion control models and frequency diversity.                                    |
|           | 1 | Physical      | Channel QoS<br>Clear Channel Assessment<br><br>Encoding<br>1/1 PN9, 1/2 convolutional Code<br><br>Rates<br>9.6 kb/s, 55.55 kb/s, 166.667 kb/s<br><br>Modulation<br>$\pm 4.8$ kHz 2-(G)FSK, $\pm 50$ kHz 2-(G)FSK or $\pm 41.667$ kHz 2-(G)FSK<br><br>Channel spacing<br>25 kHz or 200 kHz<br><br>Spectrum<br>433.060 – 434.785 MHz<br>863.000 – 870.000 MHz<br>902.000 – 928.000 MHz |

Table 4.1.1: DASH7 Alliance Protocols Stack Overview

# QUESTIONS

[info@crowdscan.be](mailto:info@crowdscan.be)

