HRIM

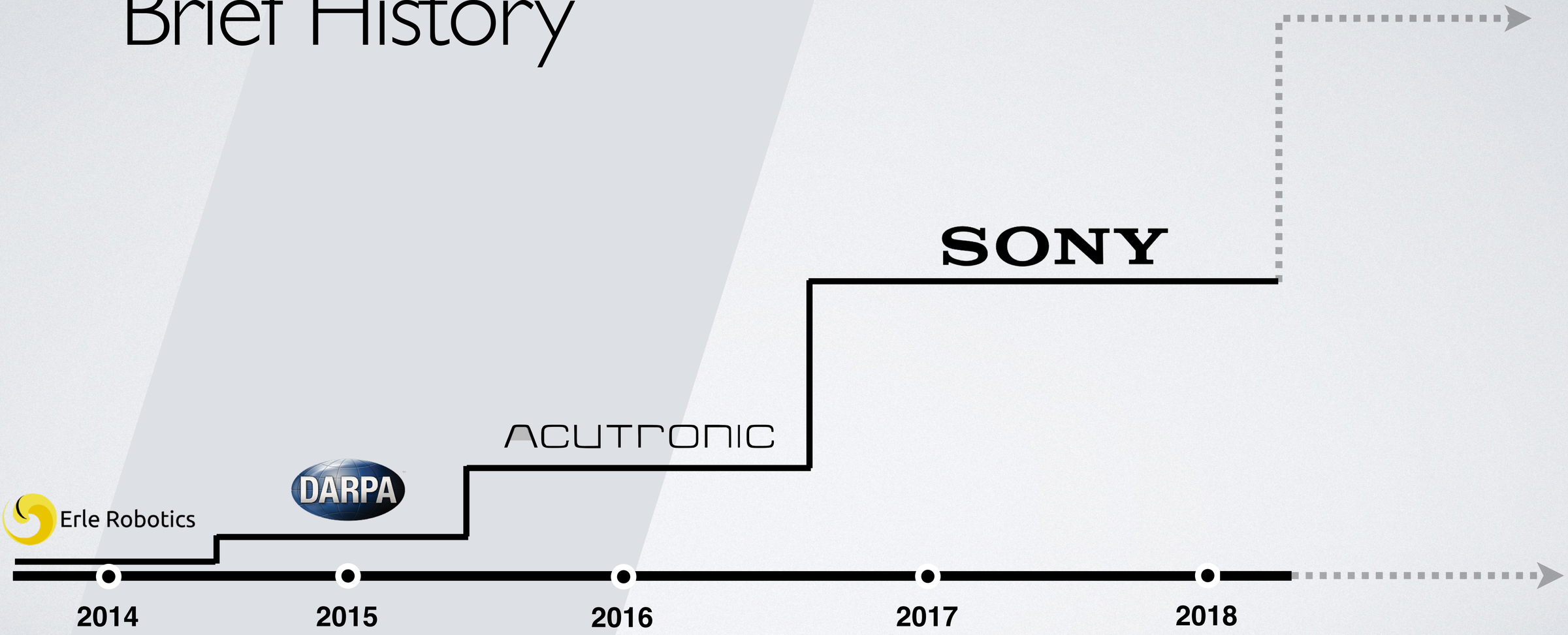
Hardware Robot Information Model

Irati Zamalloa
irati@erlerobotics.com

Erle Robotics

an Acutronic Robotics company

Erle & Acutronic Robotics:
Brief History

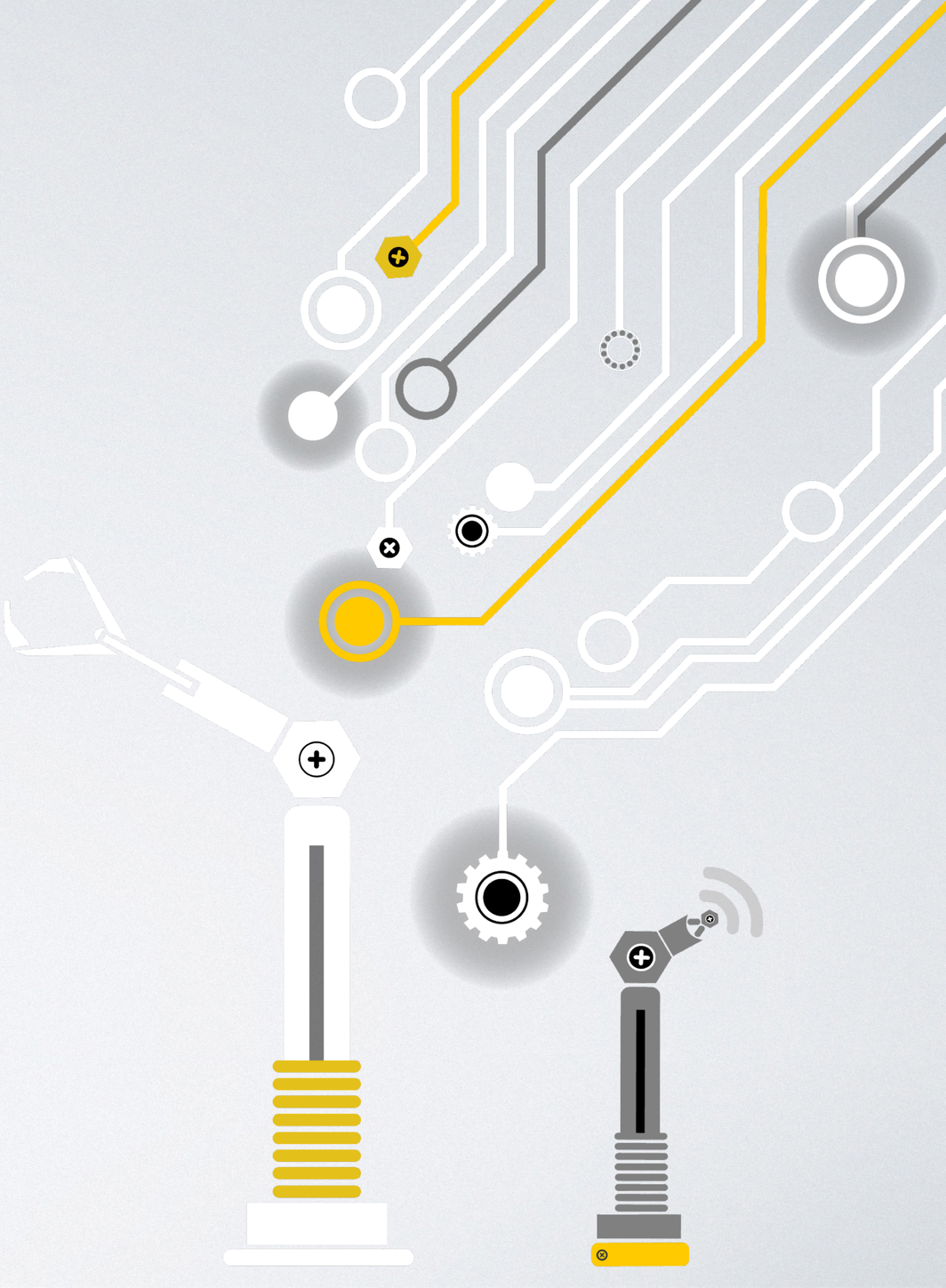


Acutronic ROBOTICS

Acutronic Robotics is a leading robotics firm focused on next-generation robot solutions around two verticals:

1. ***Modular robots, H-ROS.***

2. ***Artificial intelligence applied to robotics***.



“The more time is spent dealing with hardware/software interfaces,
the little is put into behavior development on real-world scenarios”

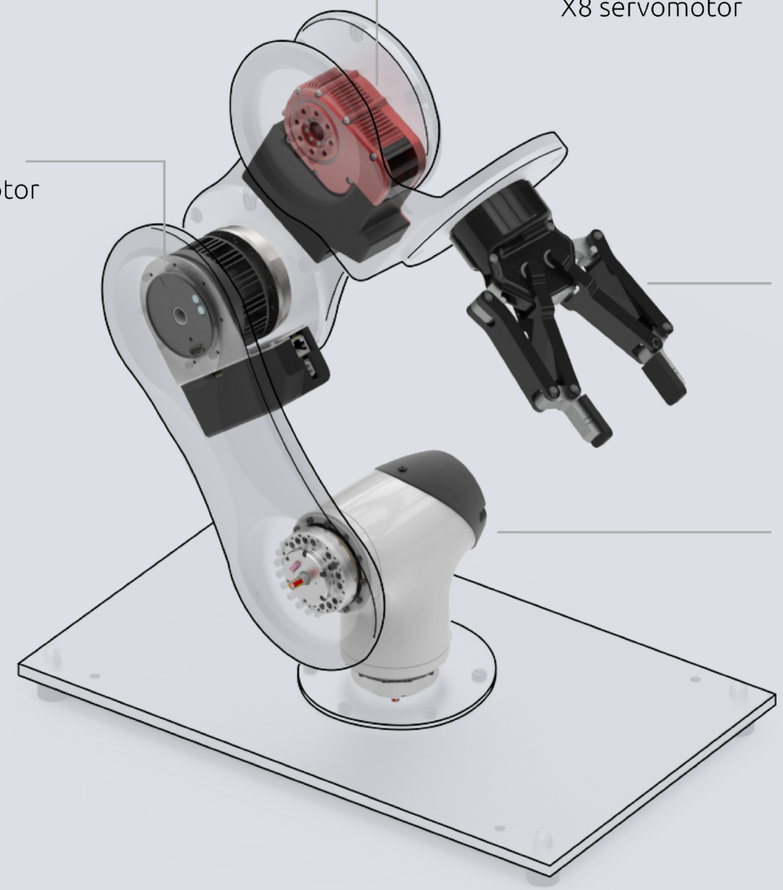
H-ROS

"A standardized software and hardware stack to easily create reusable and reconfigurable robot hardware parts."

SIMPLIFYING ROBOTICS

H-ROS

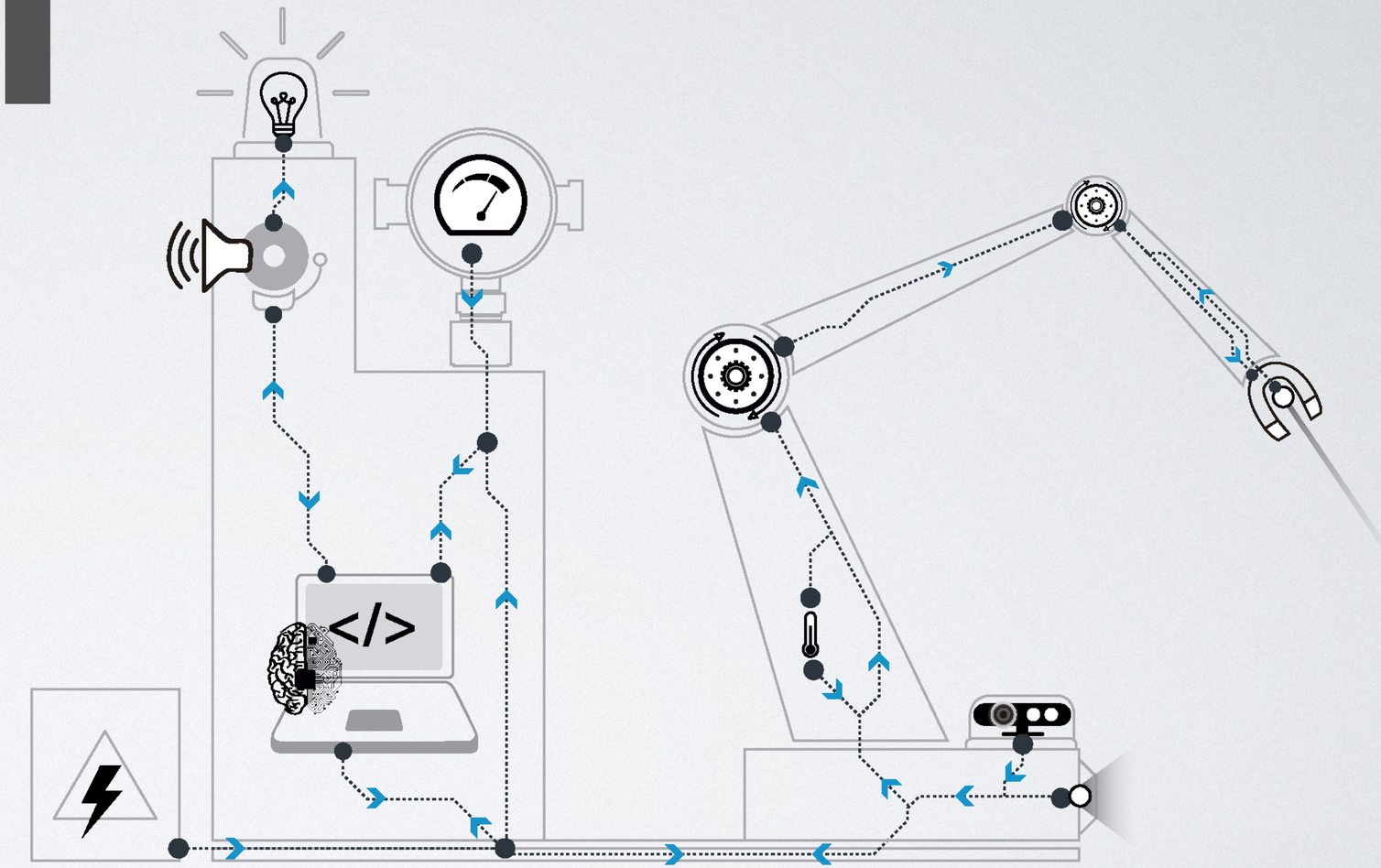
Hardware
Robot
Operating System



- Plug & Play
- Interoperable
- Extensible
- Reconfigurable
- HRIM

HRIM

"A common interface
for robot modules"



Sensor_msgs/**BatteryState.msg**

```
# Constants are chosen to match the enums in the linux kernel
# defined in include/linux/power_supply.h as of version 3.7
# The one difference is for style reasons the constants are
# all uppercase not mixed case.

# Power supply status constants
uint8 POWER_SUPPLY_STATUS_UNKNOWN = 0
uint8 POWER_SUPPLY_STATUS_CHARGING = 1
uint8 POWER_SUPPLY_STATUS_DISCHARGING = 2
uint8 POWER_SUPPLY_STATUS_NOT_CHARGING = 3
uint8 POWER_SUPPLY_STATUS_FULL = 4

# Power supply health constants
uint8 POWER_SUPPLY_HEALTH_UNKNOWN = 0
uint8 POWER_SUPPLY_HEALTH_GOOD = 1
uint8 POWER_SUPPLY_HEALTH_OVERHEAT = 2
uint8 POWER_SUPPLY_HEALTH_DEAD = 3
uint8 POWER_SUPPLY_HEALTH_OVERVOLTAGE = 4
uint8 POWER_SUPPLY_HEALTH_UNSPEC_FAILURE = 5
uint8 POWER_SUPPLY_HEALTH_COLD = 6
uint8 POWER_SUPPLY_HEALTH_WATCHDOG_TIMER_EXPIRE = 7
uint8 POWER_SUPPLY_HEALTH_SAFETY_TIMER_EXPIRE = 8

# Power supply technology (chemistry) constants
uint8 POWER_SUPPLY_TECHNOLOGY_UNKNOWN = 0
uint8 POWER_SUPPLY_TECHNOLOGY_NIMH = 1
uint8 POWER_SUPPLY_TECHNOLOGY_LION = 2
uint8 POWER_SUPPLY_TECHNOLOGY_LIPO = 3
uint8 POWER_SUPPLY_TECHNOLOGY_LIFE = 4
uint8 POWER_SUPPLY_TECHNOLOGY_NICD = 5
uint8 POWER_SUPPLY_TECHNOLOGY_LIMN = 6

Header  header
float32 voltage                 # Voltage in (V)  (Mandatory)
float32 current                 # Negative when discharging (A)  (If unmeasured NaN)
float32 charge                  # Current charge in Ah  (If unmeasured NaN)
float32 capacity                # Capacity in Ah (last full capacity)  (If unmeasured NaN)
float32 design_capacity         # Capacity in Ah (design capacity)  (If unmeasured NaN)
float32 percentage              # Charge percentage on 0 to 1 range  (If unmeasured NaN)
uint8   power_supply_status     # The charging status as reported. Values defined above
uint8   power_supply_health     # The battery health metric. Values defined above
uint8   power_supply_technology # The battery chemistry. Values defined above
bool    present                 # True if the battery is present

float32[] cell_voltage          # An array of individual cell voltages for each cell in the pack
                                # If individual voltages unknown but number of cells known set
                                # each to NaN
string location                 # The location into which the battery is inserted)
string serial_number            # The best approximation of the battery serial number
```

Cob_msgs/**PowerState.msg**

```
# This message communicates the state of the power system.
Header header
float64 voltage                 # [V]
float64 current                 # [A]
float64 power_consumption       # [W] can only be calculated if not charging
float64 remaining_capacity      # [Ah]
float64 relative_remaining_capacity  # [0..100] percent of maximum capacity (parameter max_capacity)
bool charging                   # flag if robot is connected to external power or not
float64 time_remaining          # [h] estimated time to empty or fully charged
float64 temperature             # [Celsius] temperature of the battery
```

Mavros_msgs/**BatteryStatus.msg**

```
std_msgs/Header header
float32 voltage
float32 current
float32 remaining
```

Kobiki_msgs/**PowerSystemEvent.msg**

```
uint8 UNPLUGGED          = 0
uint8 PLUGGED_TO_ADAPTER  = 1
uint8 PLUGGED_TO_DOCKBASE = 2
uint8 CHARGE_COMPLETED    = 3
uint8 BATTERY_LOW         = 4
uint8 BATTERY_CRITICAL    = 5

uint8 event
```

Naoqi (http://docs.ros.org/jade/api/naoqi_bridge_msgs/html/msg/Bumper.html)	**Kobuki** (http://docs.ros.org/hydro/api/kobuki_msgs/html/msg/BumperEvent.html)	**Evarobot** http://wiki.ros.org/evarobot_bumper
Publisher: **bumper**: uint8 bumper # which bumper (left or right) uint8 state # state of the bumper (pressed or released) uint8 right=0 uint8 left=1 uint8 back=2 uint8 stateReleased=0 uint8 statePressed=1	**Publisher:** **bumper**: uint8 LEFT = 0 uint8 CENTER = 1 uint8 RIGHT = 2 uint8 RELEASED = 0 uint8 PRESSED = 1 uint8 bumper uint8 state	**Publisher:** **bumper**: std_msgs/Header header bool [] state
		Parameters: ~i2c_path (string, default: /dev/i2c-1) ~commandTopic (string, default: bumper) ~frequency (double)

HRIM

The Hardware Robot Information Model

Interoperability

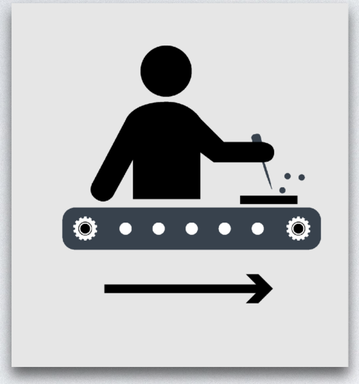
Modularity

Solid infrastructure

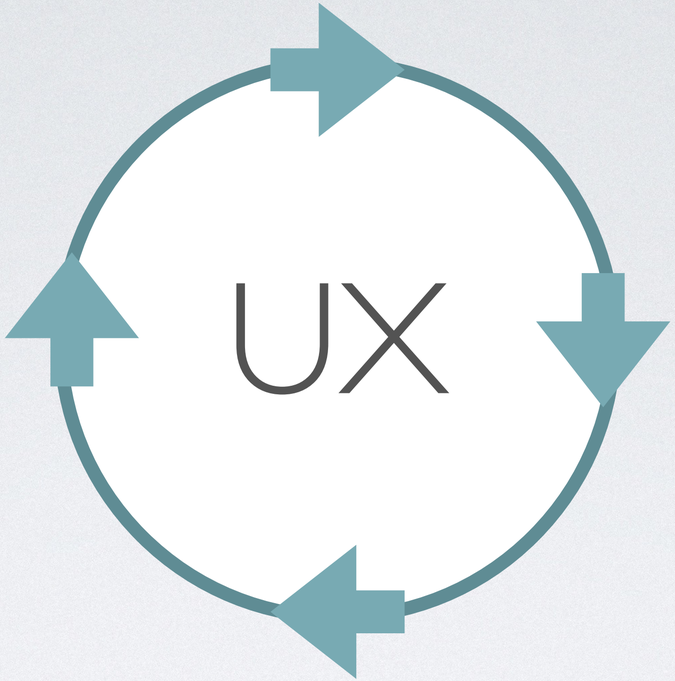
Collaborative

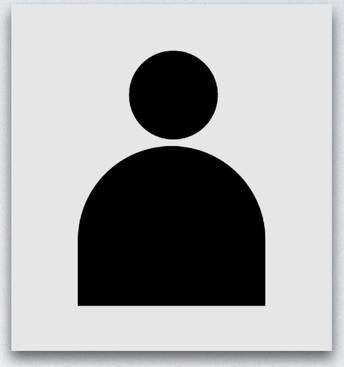
github.com/erlerobot/HRIM

ROS 2.0



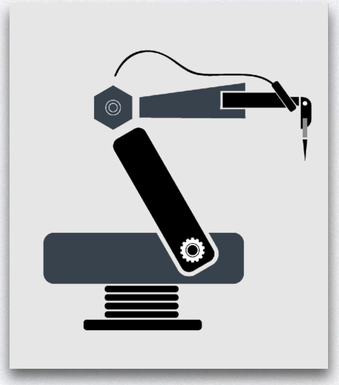
MANUFACTURERS





USERS

DEVELOPERS



INDUSTRY

CLASSIFICATION

Cognition

Actuator
Rotary servomotor
Speaker
...

Sensor
Camera
Range finder
...

Communication
WiFi
Switch
...

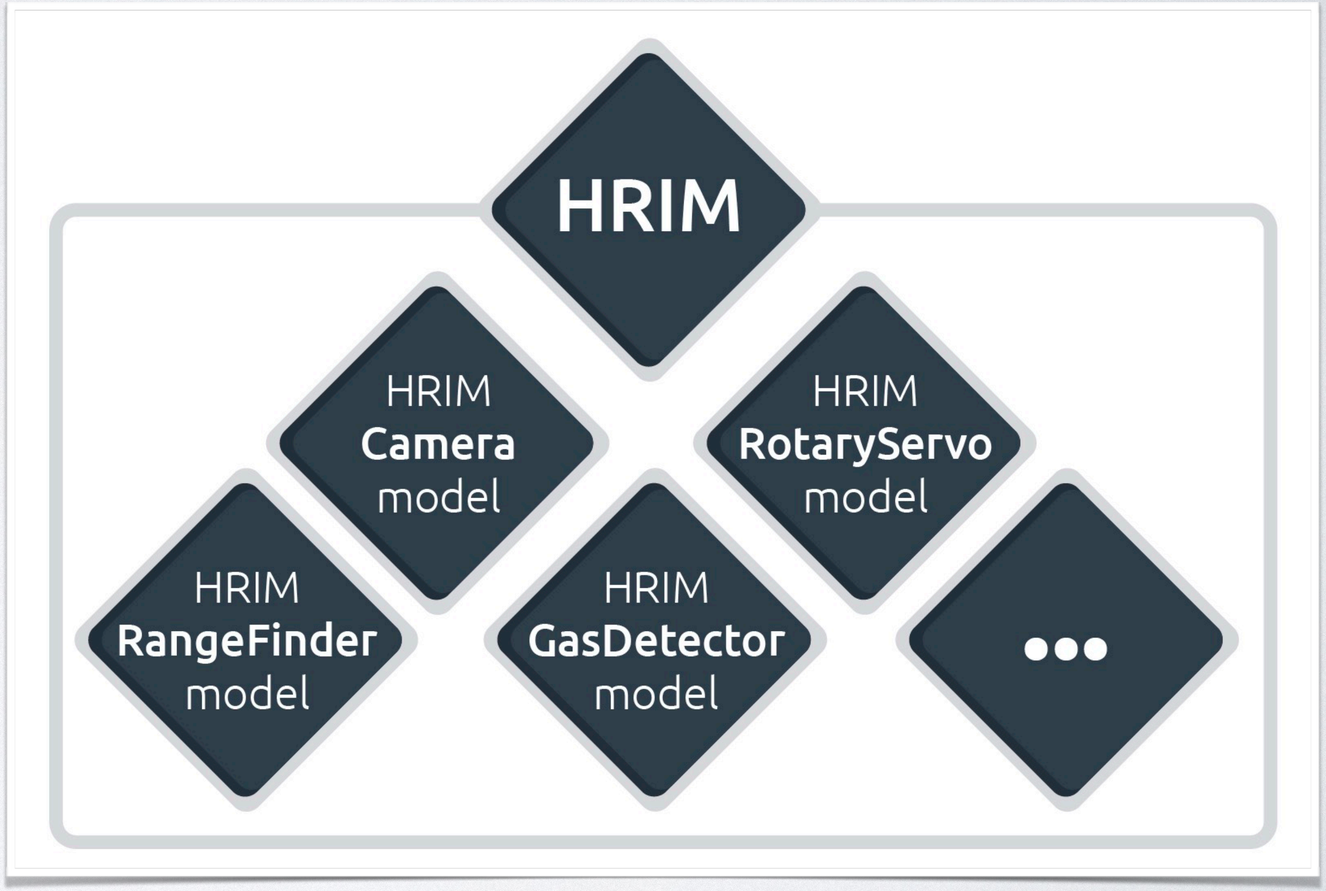
User Interface
Joystic
Tactile screen
...

Power
Battery
Power supply
...

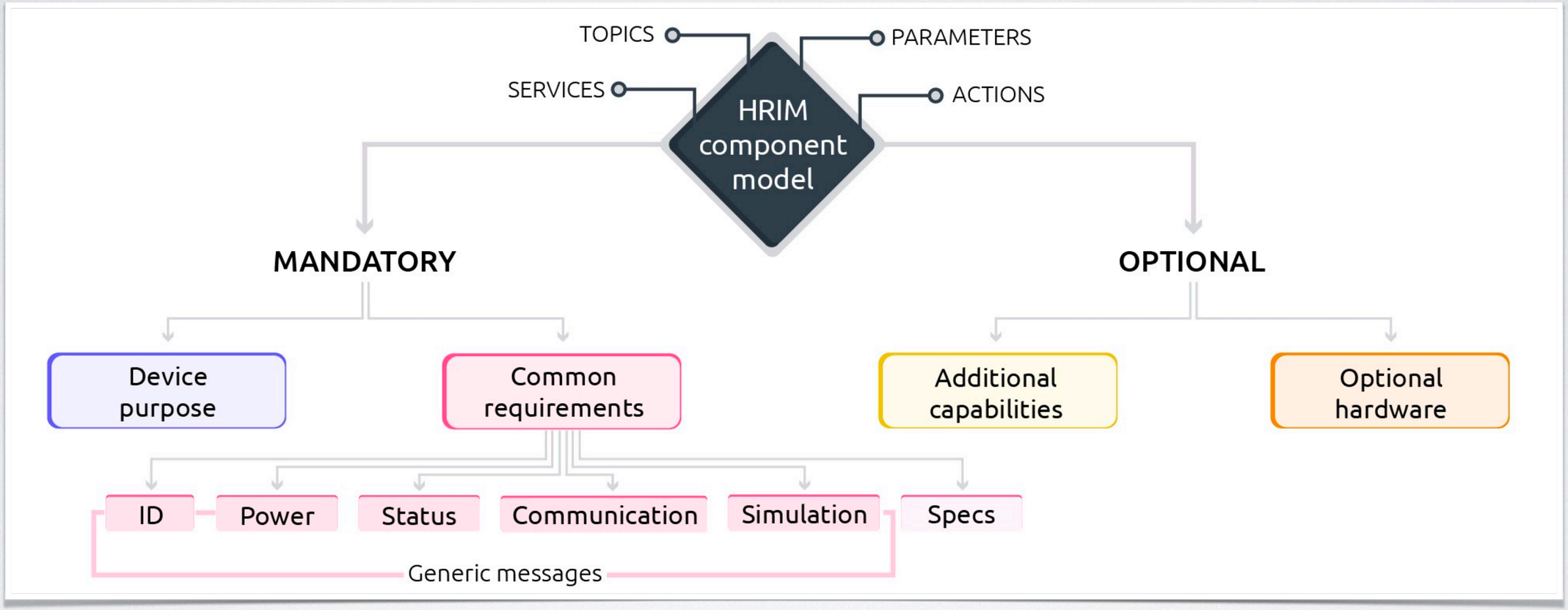
Composite
Mobile base
Conveyor
...

HRIM <component> Model

- Most of robotics components.
- Designed one by one:
  - Analysis
  - Conclusion
  - Create the model
  - Improve
- Updates.



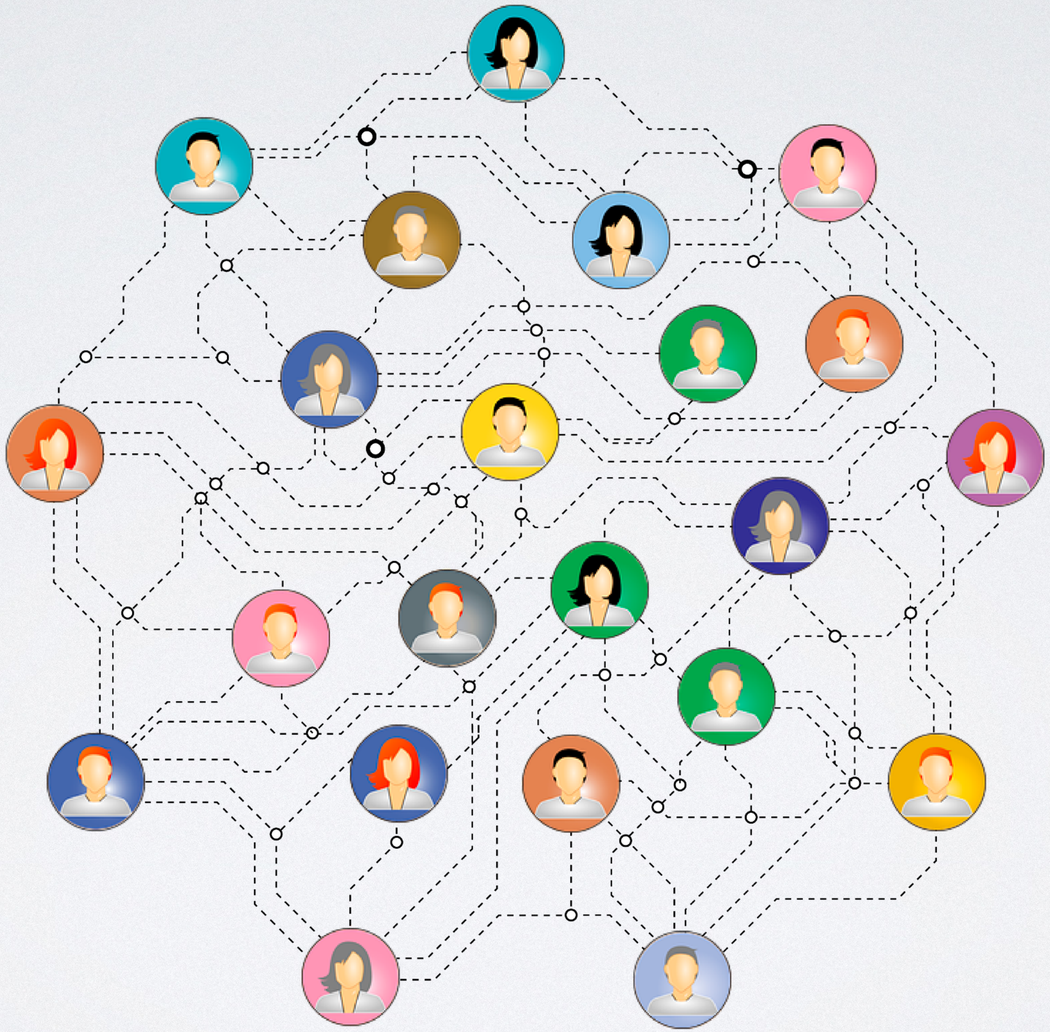
General structure



The general structure in which all the HRIM component models are based on. Each component has *topics*, *services*, *parameters* and *actions* to communicate. For each one of these abstractions, the figure illustrates that some will be mandatory and some others optional.

docs.h-ros.com/hrim

COLLABORATION



CONCLUSION

- Robotics community need a common interface database focused on hardware.
- HRIM offers to the robotics community a common interface that facilitates the manufacturing of reusable and interoperable robot hardware module.
- HRIM it is being built side by side with manufacturers and experts.
- All we win.

NEXT STEPS

- Packages generator.
- MDE techniques.
- FTP.
- Electronic datasheet.

THANKS!!

Erle Robotics

an Acutronic Robotics company

Irati Zamalloa
irati@erlerobotics.com

Alejandro Hernández
alex@erlerobotics.com