



Yet another level 1 trigger sorter for LHCb



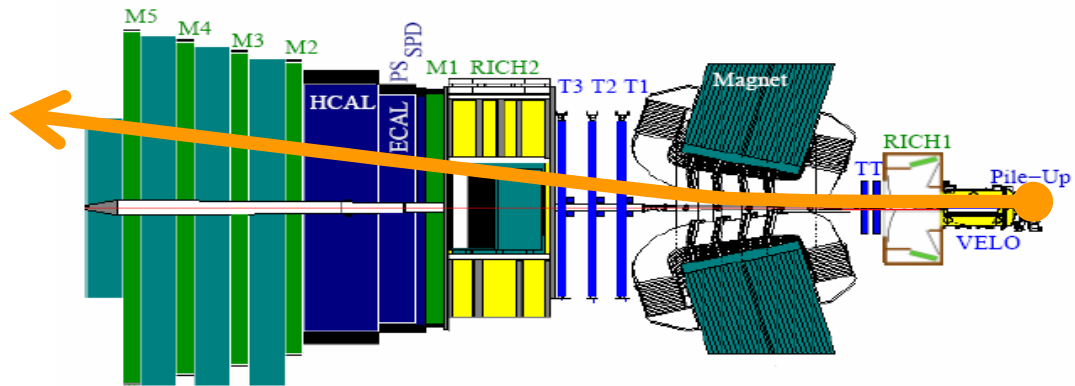


Contents

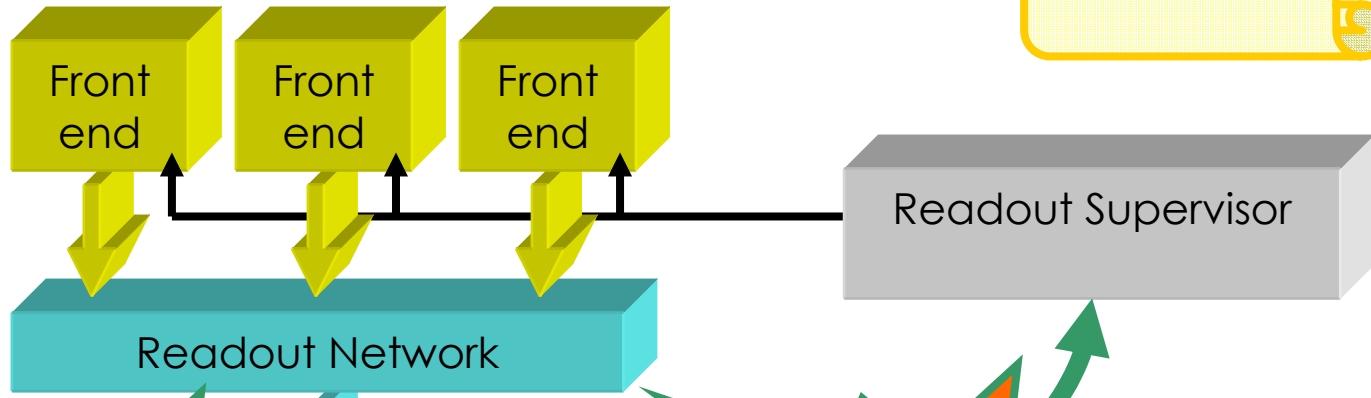
- ☀ Background
- ☀ Design solution
- ☀ Tests and results



Background



Where is the problem?



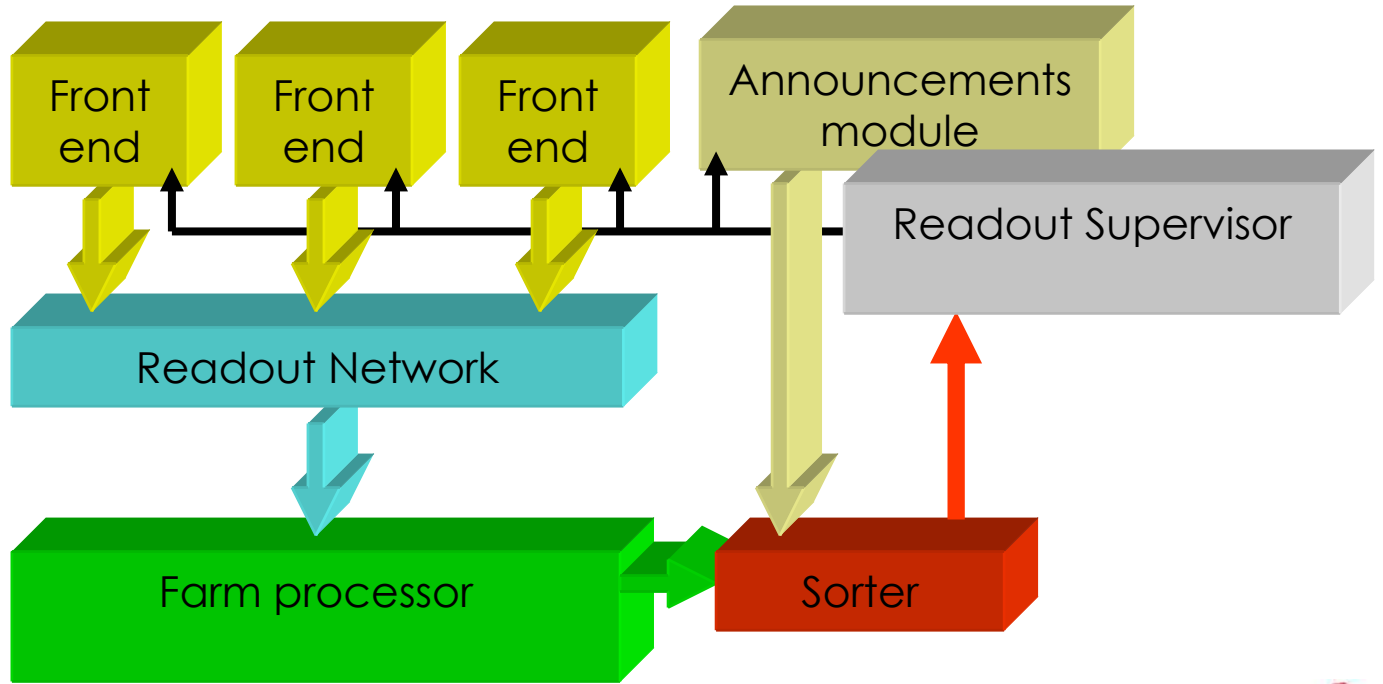
Random time of processing

Sequential order is lost



Background

So, we need a sorter





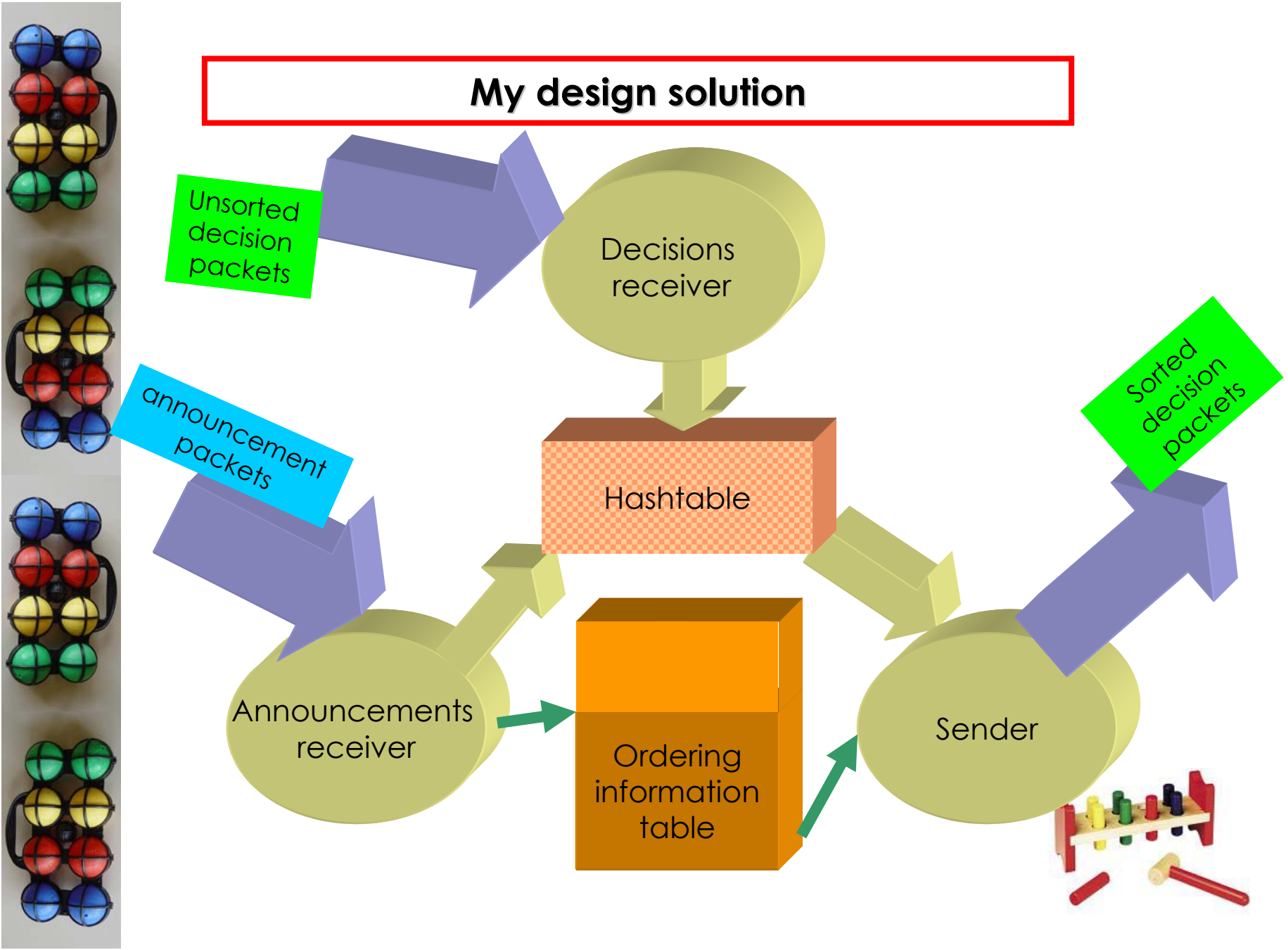
What I have to do ?

I have to implement a sorter
in software on just one multiprocessor PC

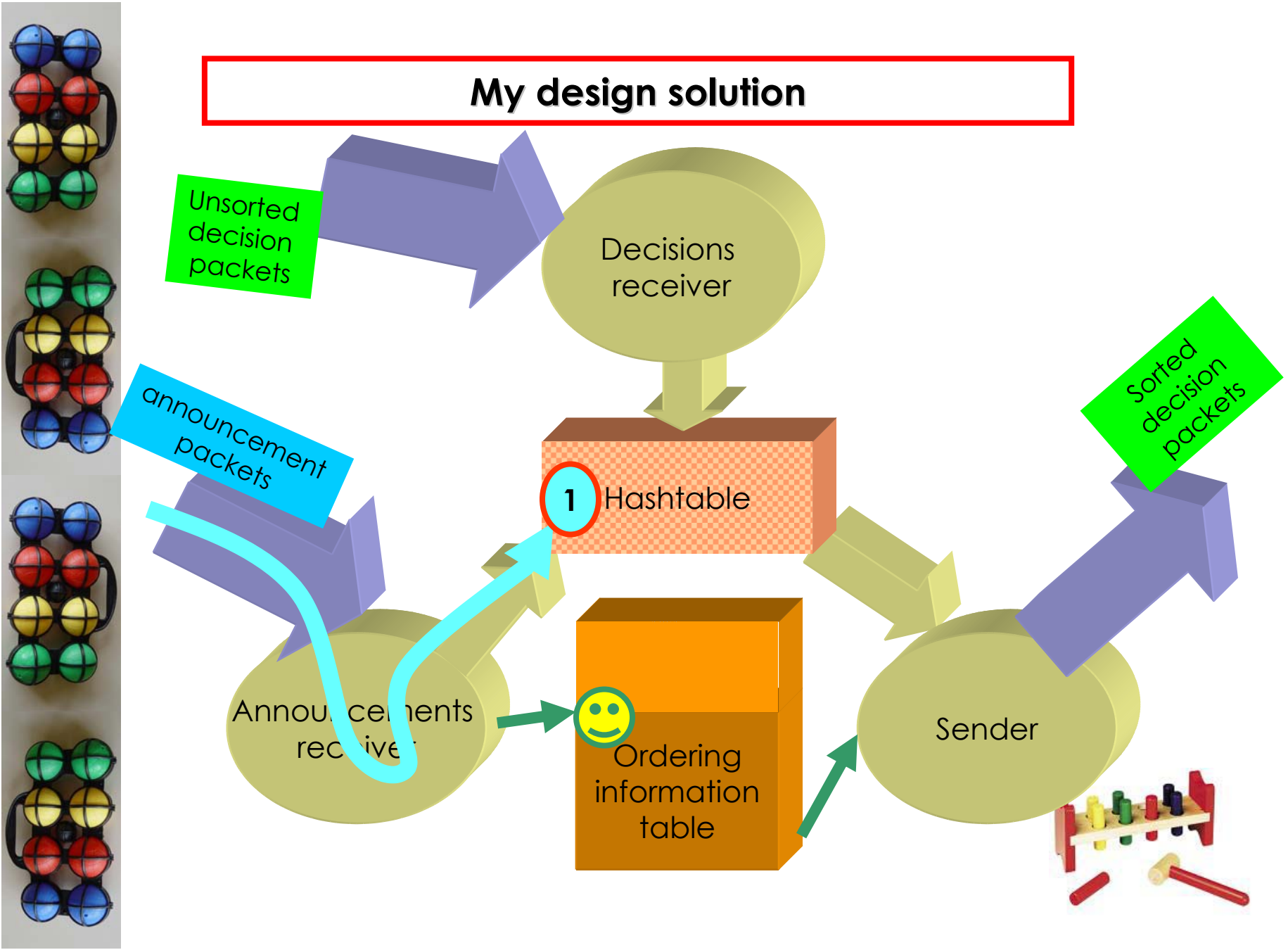
- ✓ with a packet rate of 40 kHz for decision event
- ✓ With detection of missing packet



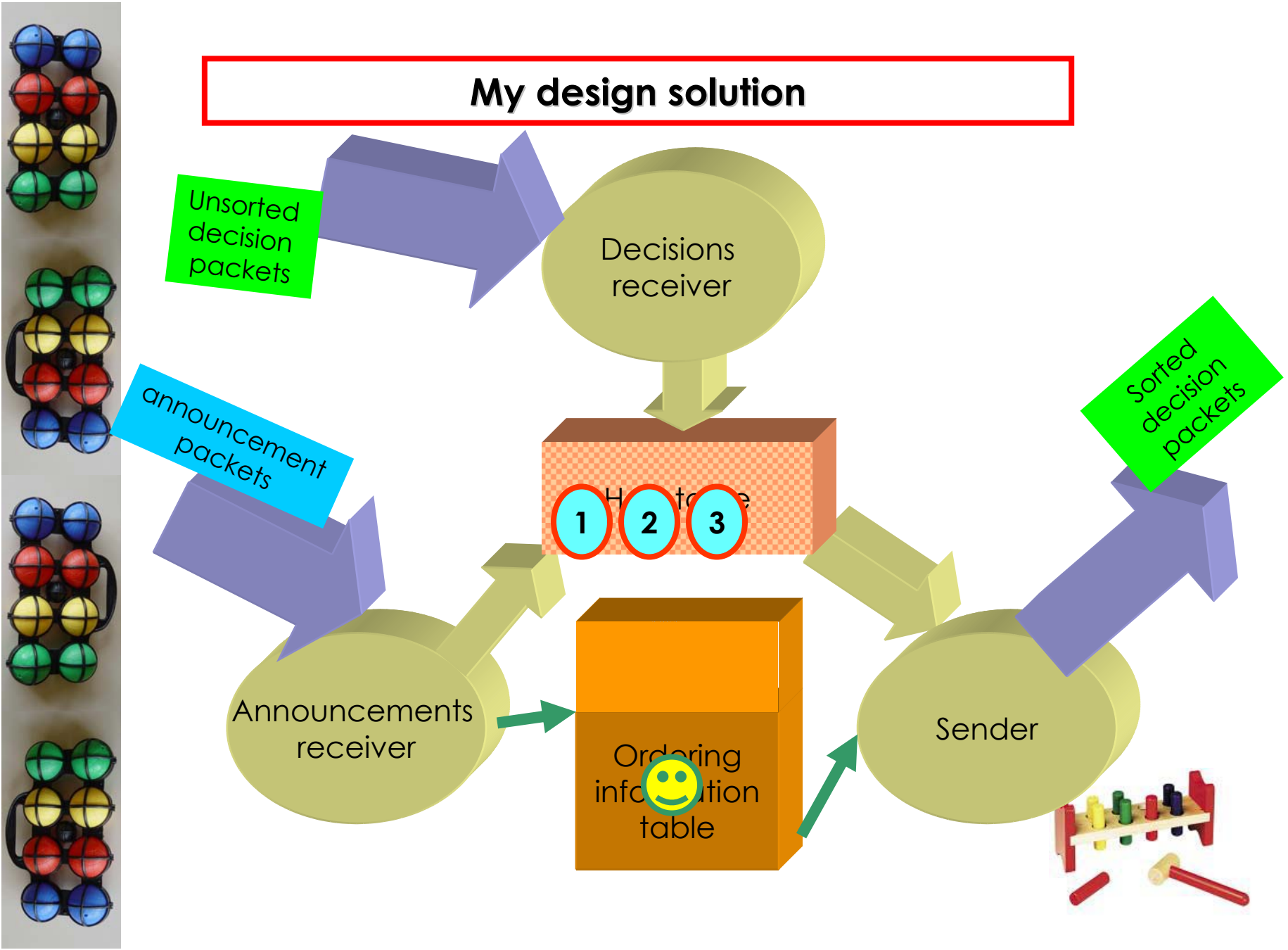
My design solution



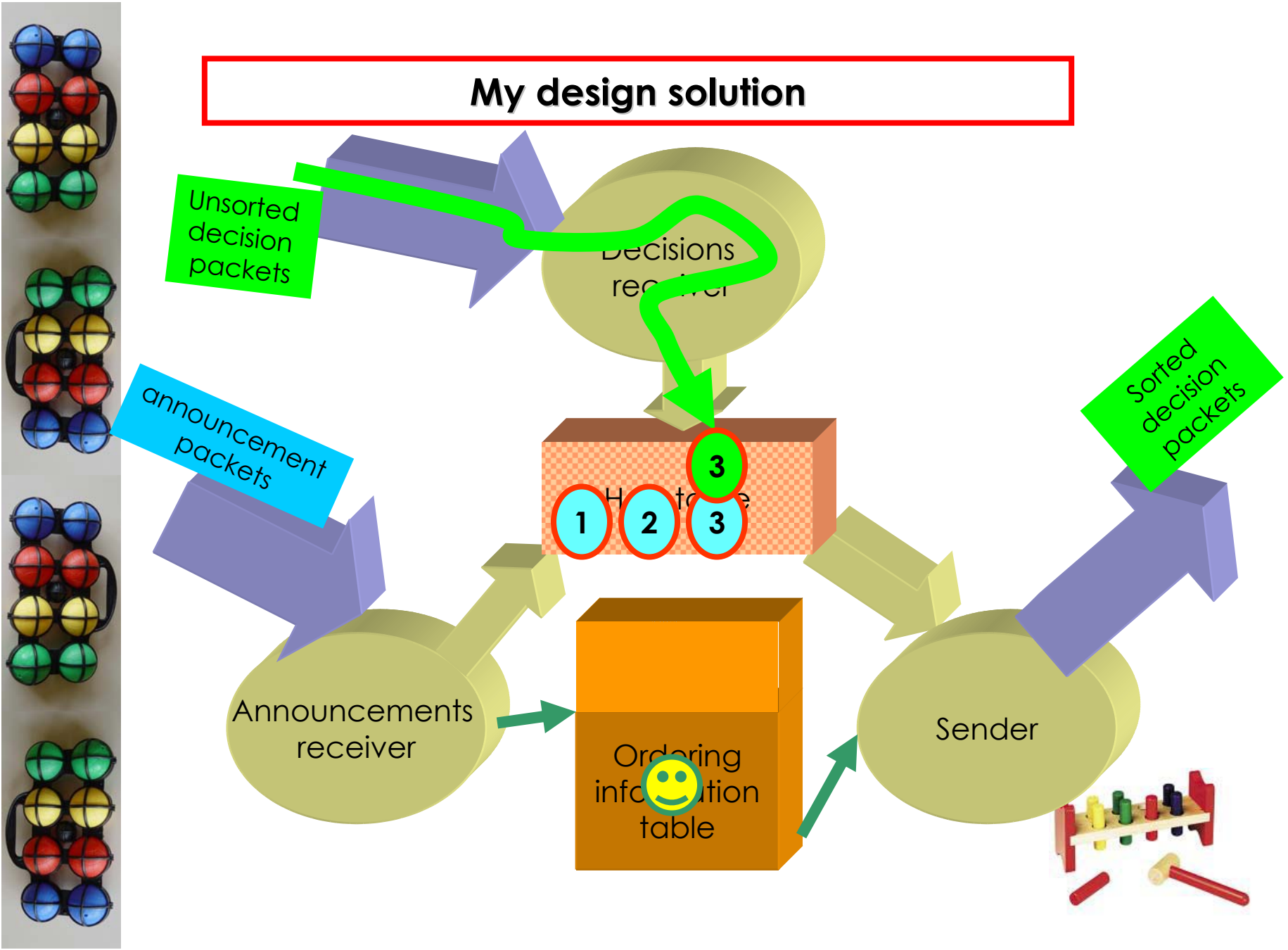
My design solution



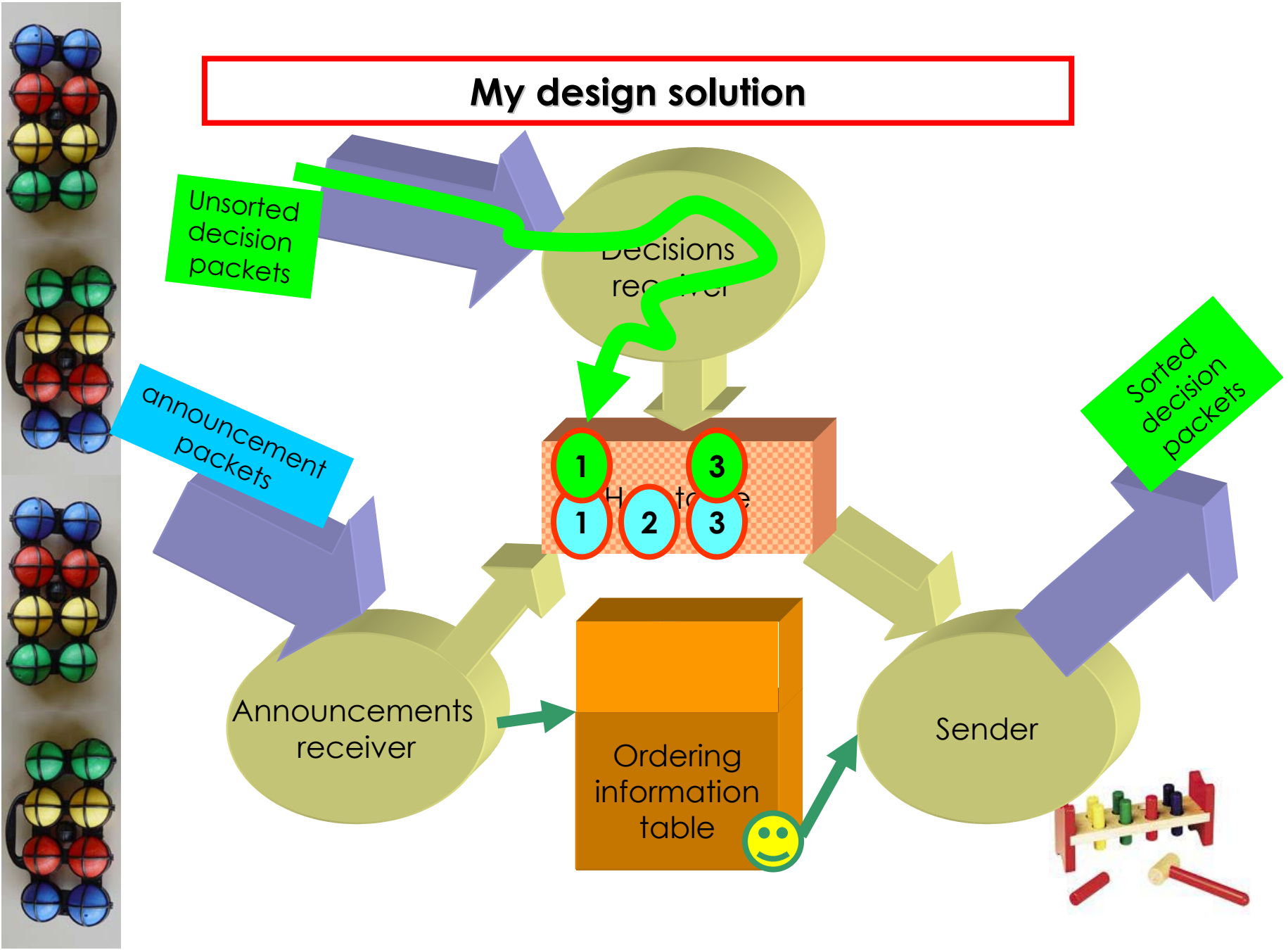
My design solution



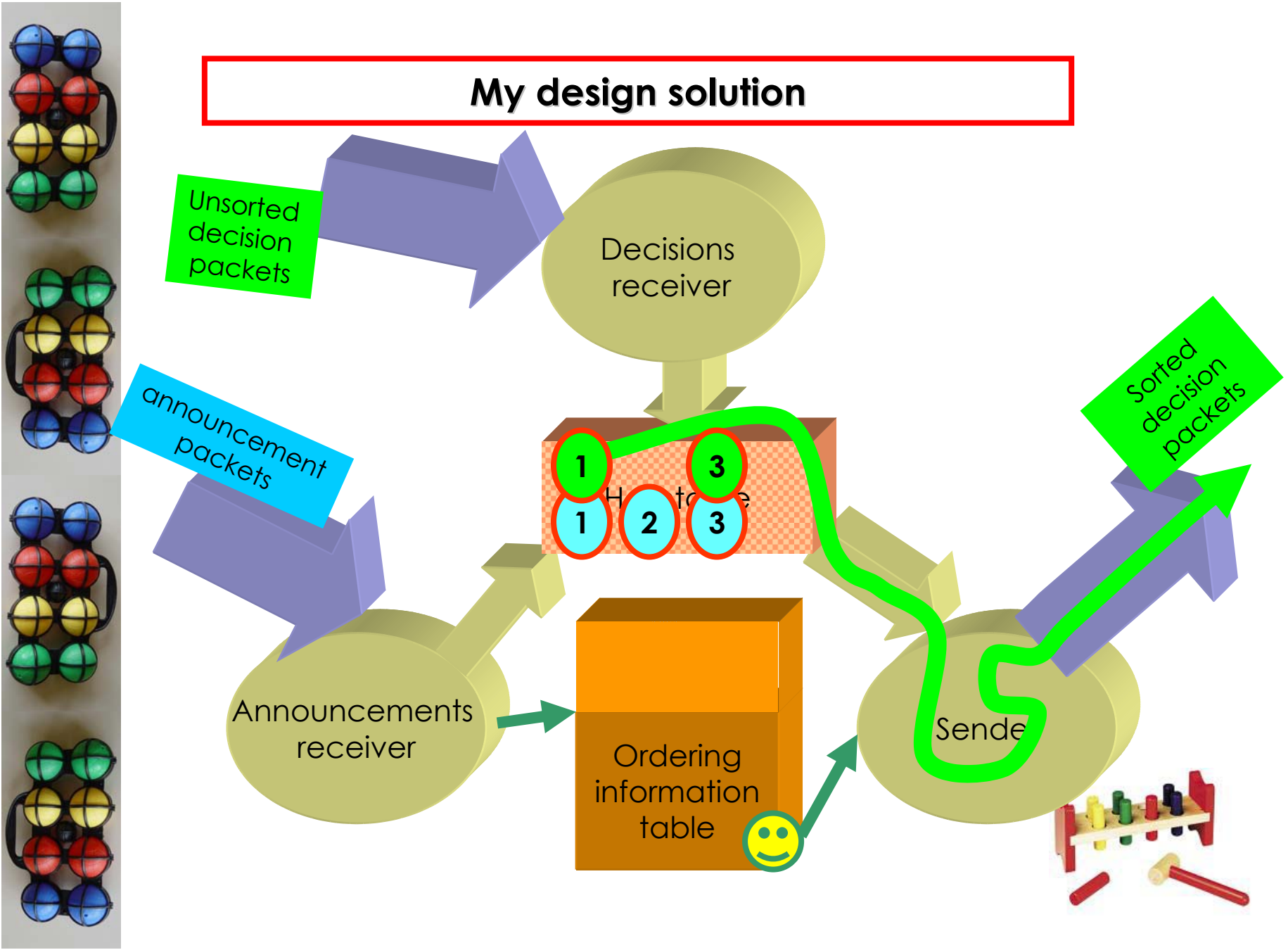
My design solution



My design solution



My design solution





Design solution

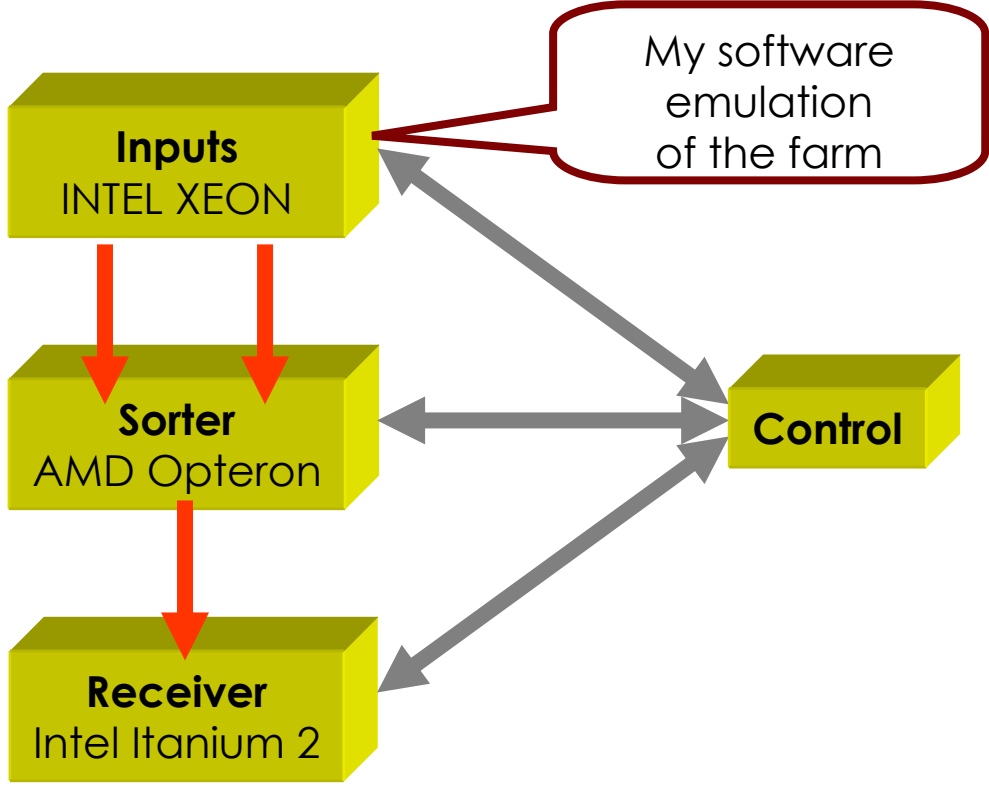
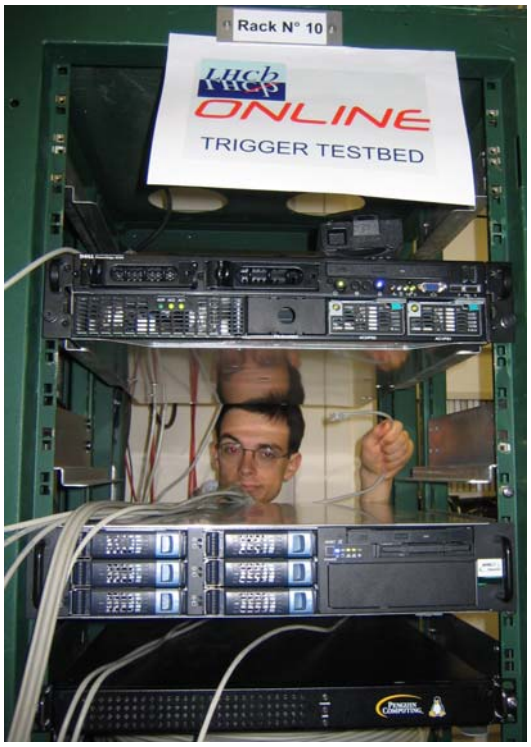
This design allows :

- ✓ A fast sorting
- ✓ Robustness against lost and unexpected events.





Tests



Linux kernel 2.6.6

More than 10^{10}
multidecisions packets
transferred



Tests and results

A first naïve implementation can run at the nominal frequency of 40kHz, enough during an afternoon...



I improve the robustness by :

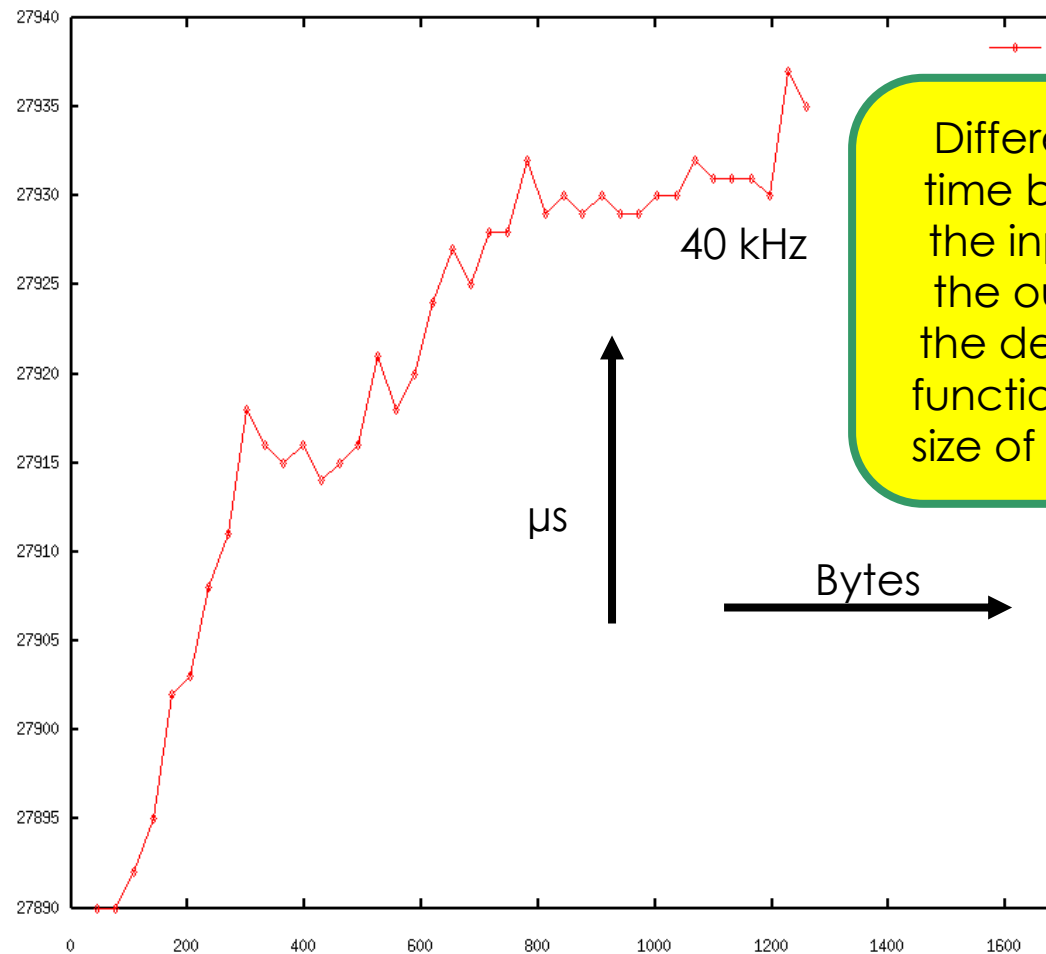
- ✓ tuning a lot of parameters in the Linux kernel
- ✓ using some atypical programming techniques





Results

In my simulation, a decision packet is send to the sorter with a latency between 1ms and 60 ms (uniform probability) after an announcement.

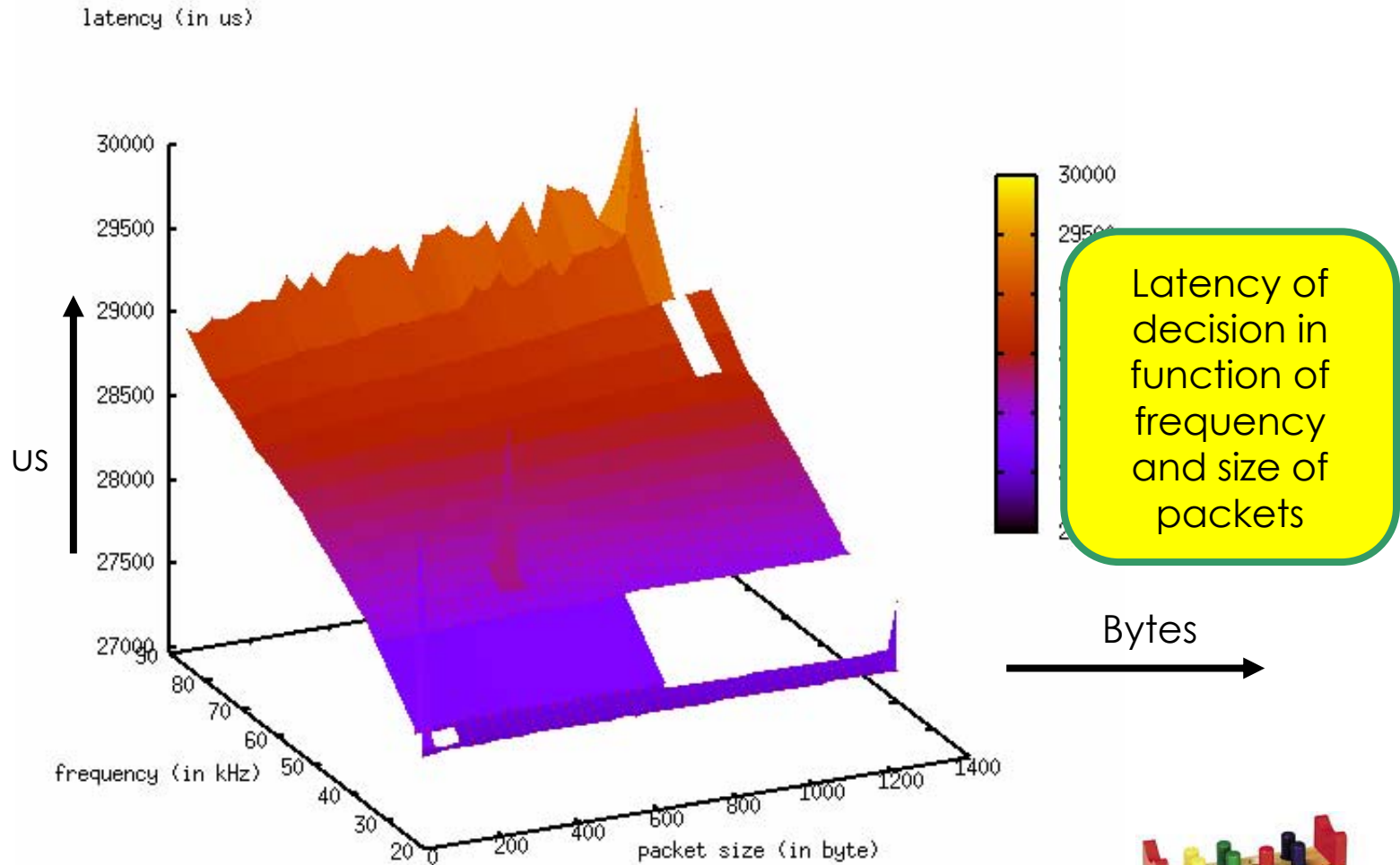


Difference of time between the input and the output of the decision in function of the size of packets





Results



view: 67.0000, 335.000 scale: 1.00000, 1.00000



Results

Not too bad... but,

My program for the inputs is less efficient than the sorter...
to go further, I will use real electronic module
to emulate the farm processor



Questions ?

