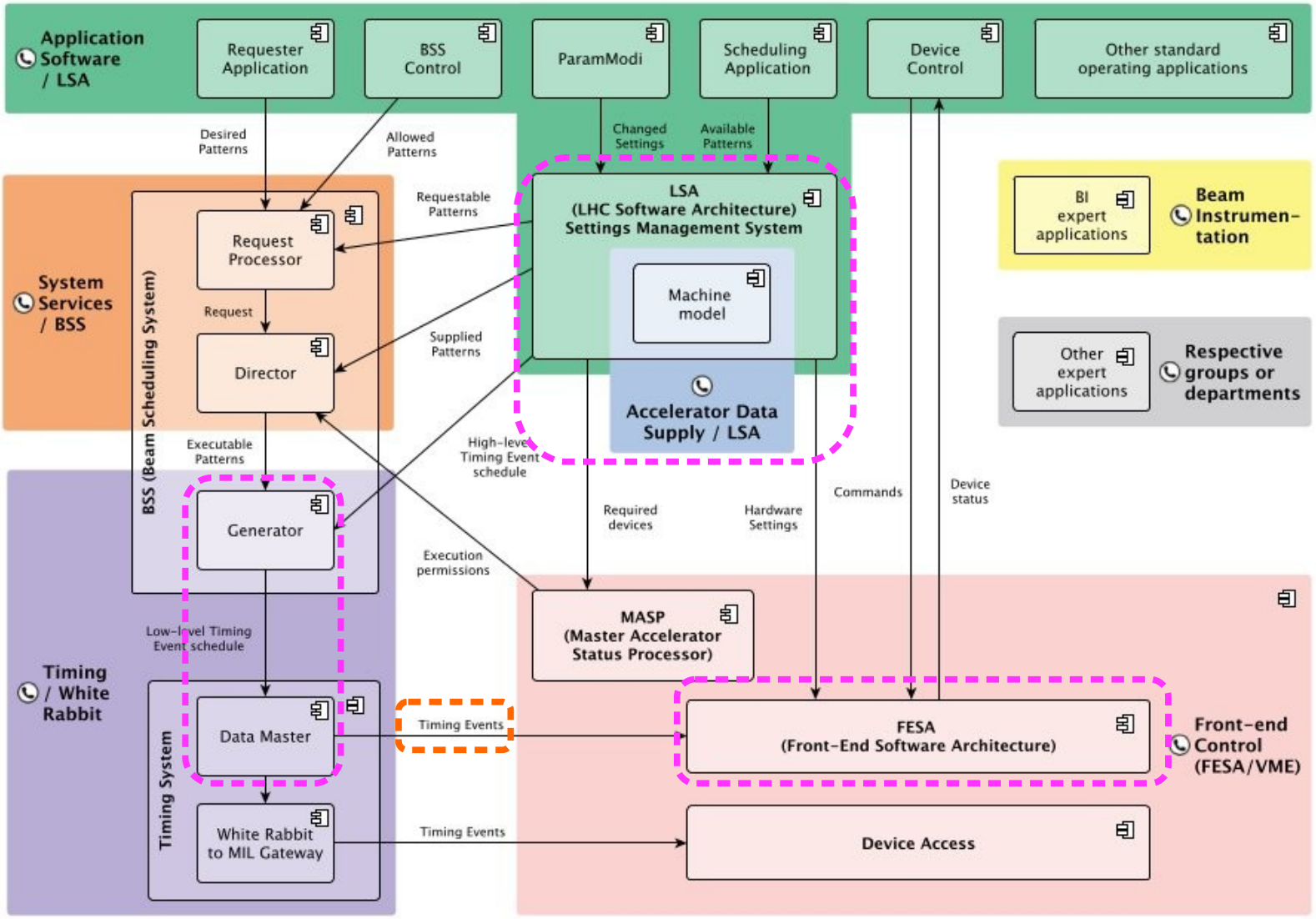
# Who you gonna call?




**What is this?**
**What is it good for?**

This is an simplified view on (a part of) the control system's architecture, created with the intention to help you make an educated guess on who to call when something's not working. If you're not sure, don't worry. It'll take time to get to know the new control system structures and no one will get mad if you call the "wrong" group.

Please be aware that the diagram is focused on certain areas of the control system and consequently, other equally important components are missing. Also, consider this diagram to be work-in-progress. If you'd like to contribute, see below.

**What do the symbols mean?**

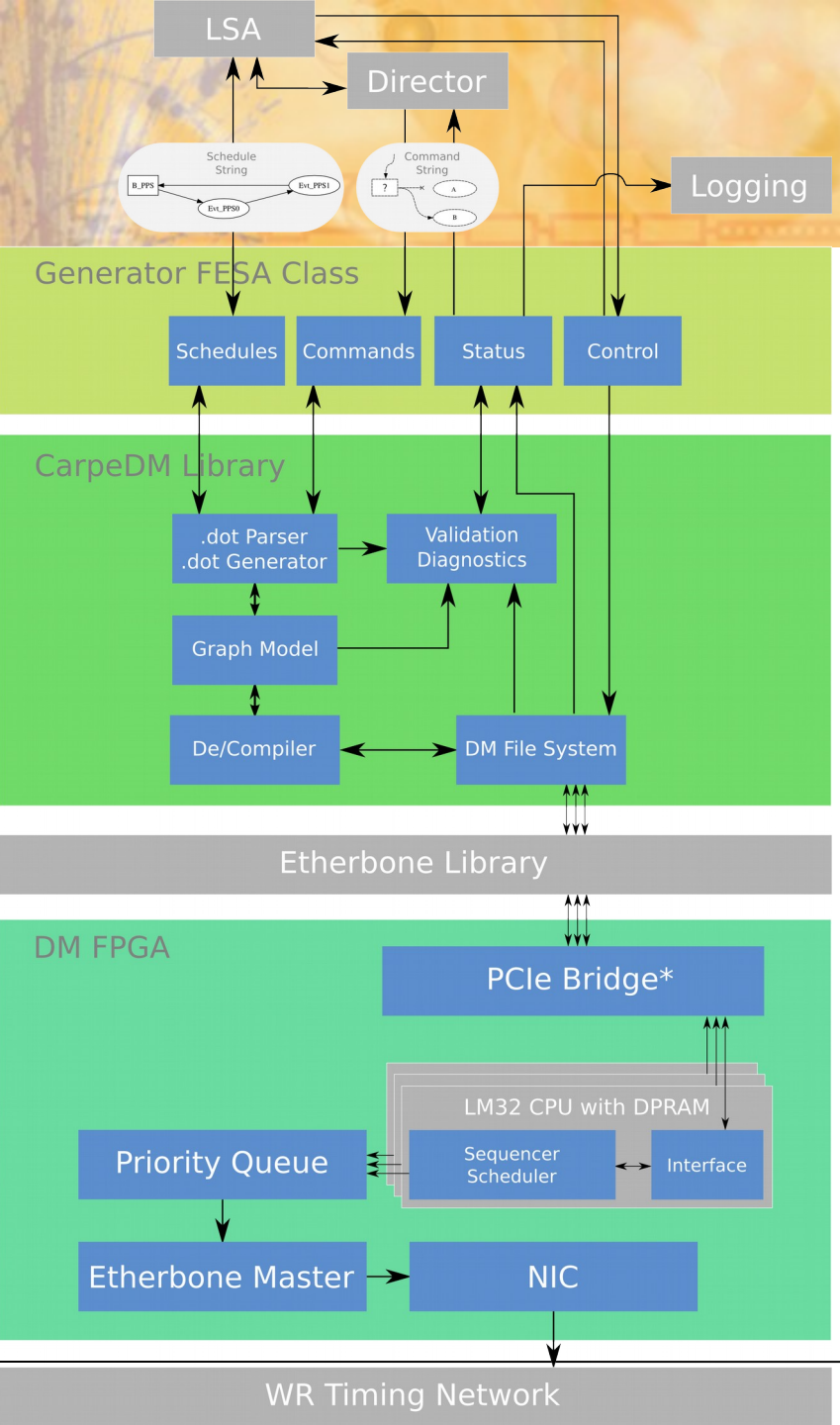
The boxes symbolize applications, components or subsystems of the control system. The arrows stand for data flows between them. The colored regions represent areas of responsibility. The terms next to the telephone icons are taken from FSN (when switched to English) and may help you look up the on-call number you need to dial.

**Who can I ask about it?**

If you have any questions, comments, suggestions or corrections regarding this diagram, please feel free to call Hanno at -3089 or write to h.huether@gsi.de.

**Thanks!**

# Primer: Data Master



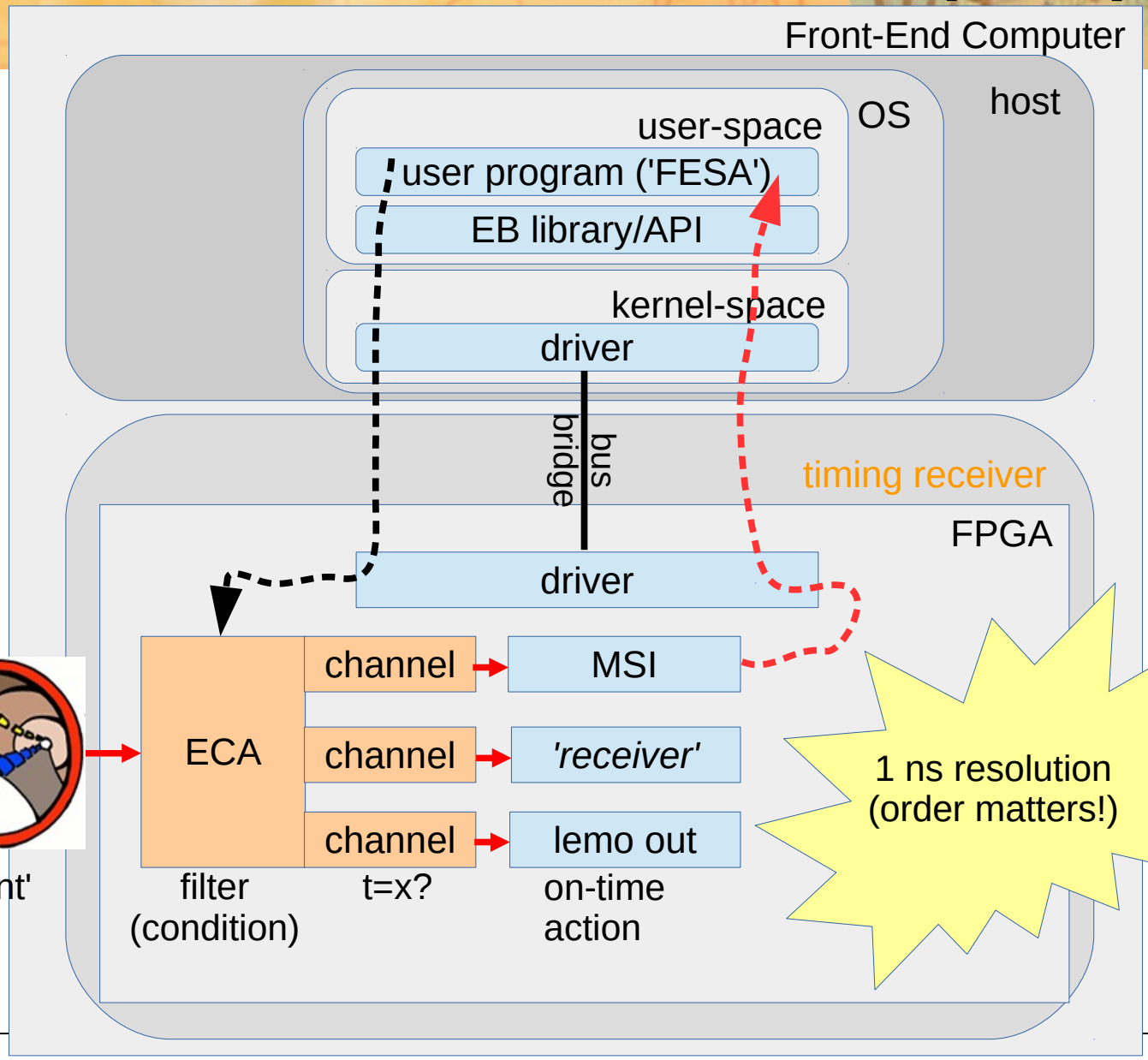

- LSA generates so-called 'patterns groups'
- Director controls pattern execution by commands
- Generator FESA class as interface to DM
- CarpeDM library as interface to firmware
- DM FPGA
  - runs on standard PCIe Timing Receiver (ArriaV)
  - lm32 cluster, priority Q, ~~ECA~~

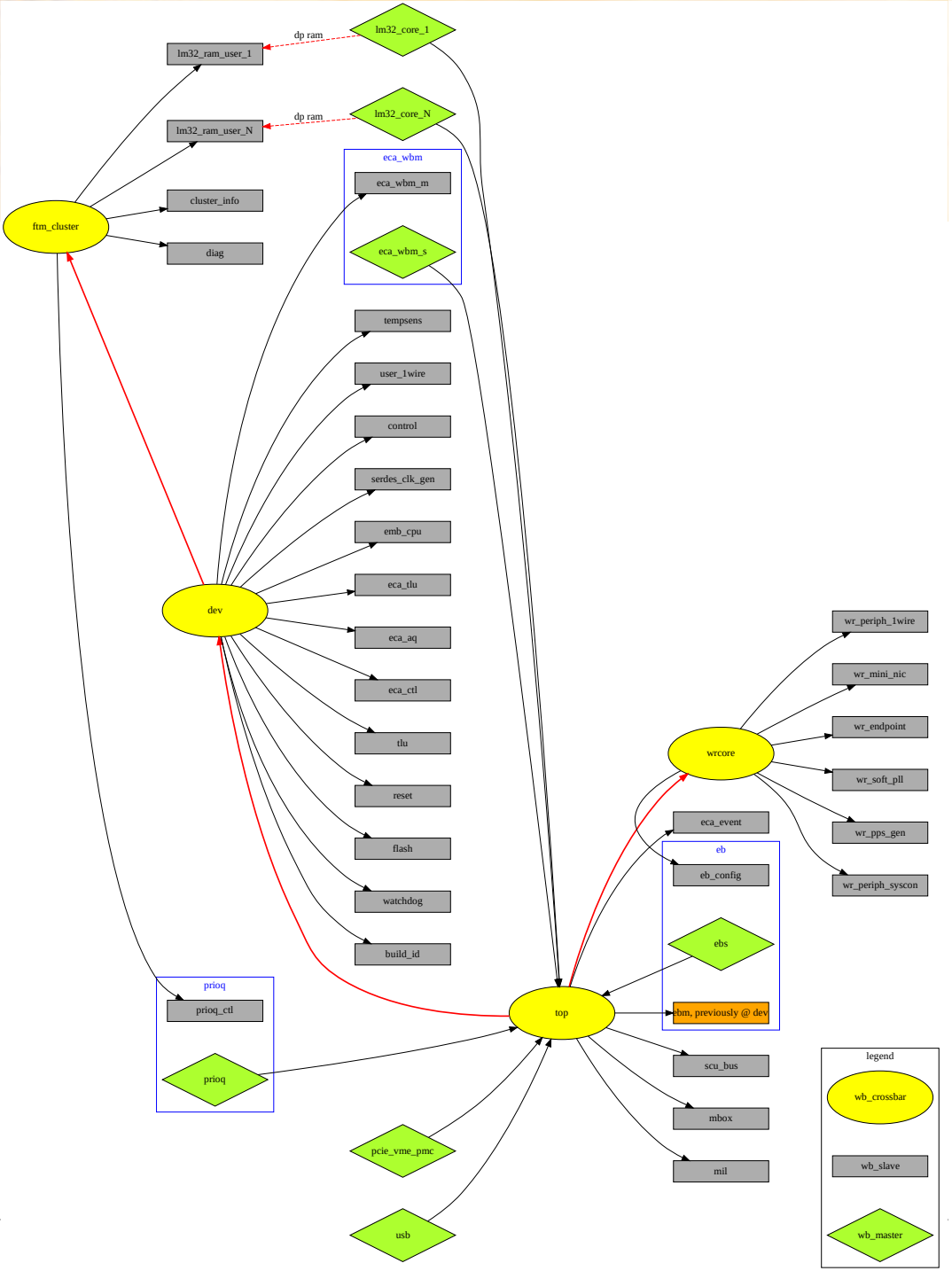
DM starts scheduling telegrams 1ms ahead of deadline.

- 500us margin for DM
- 500us margin for WR network

# Primer: Event Condition Action Unit (ECA)

- Event: DM telegram
- Condition: index
- Action: configured, executed on-time






## FPGA

- WR (OHWR) + Timing/Control (GSI)
- ~ 380k lines of VHDL
- > 30 IP cores + lm32 softcores
- on-chip Wishbone bus (with clock crossings)

include lot's of functionality considered too complex to be solved in software :-O

## Control

SCU backplane bus to slave/interface boards

MIL device bus still very heavily used

function generator example, control SIS dipoles: from FESA → lm32 → SCU backplane → slave boards → MIL bus → 6 (!) interface boards with drifting clocks operated outside specs → on-the-fly-supply with new values every 2ms

## Weihnachtswunschzettel 'E' Release

- new WR core 4.2
- new WRS 5.1
- new PCIe driver
- new saftlib
- support new DAQ @ 1kHz
- migration Quartus 16 → Quartus 18.1
- solution to RAM issues
- fixing on-chip timing issues
- ...