

DABC for HADES



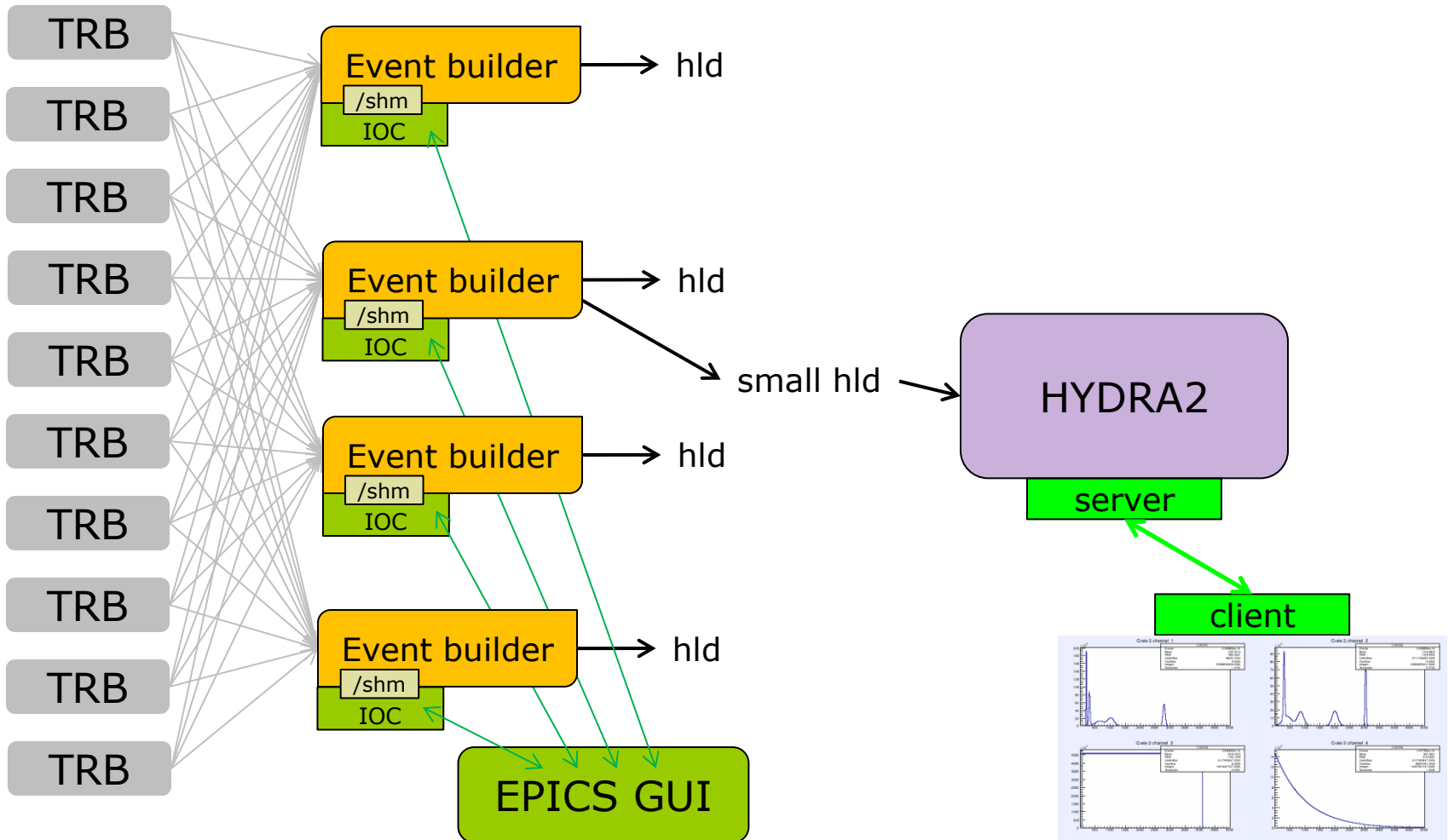
Sergey Linev,
Jörn Adamczewski-Musch

GSI / Experiment Electronic
27.10.2014

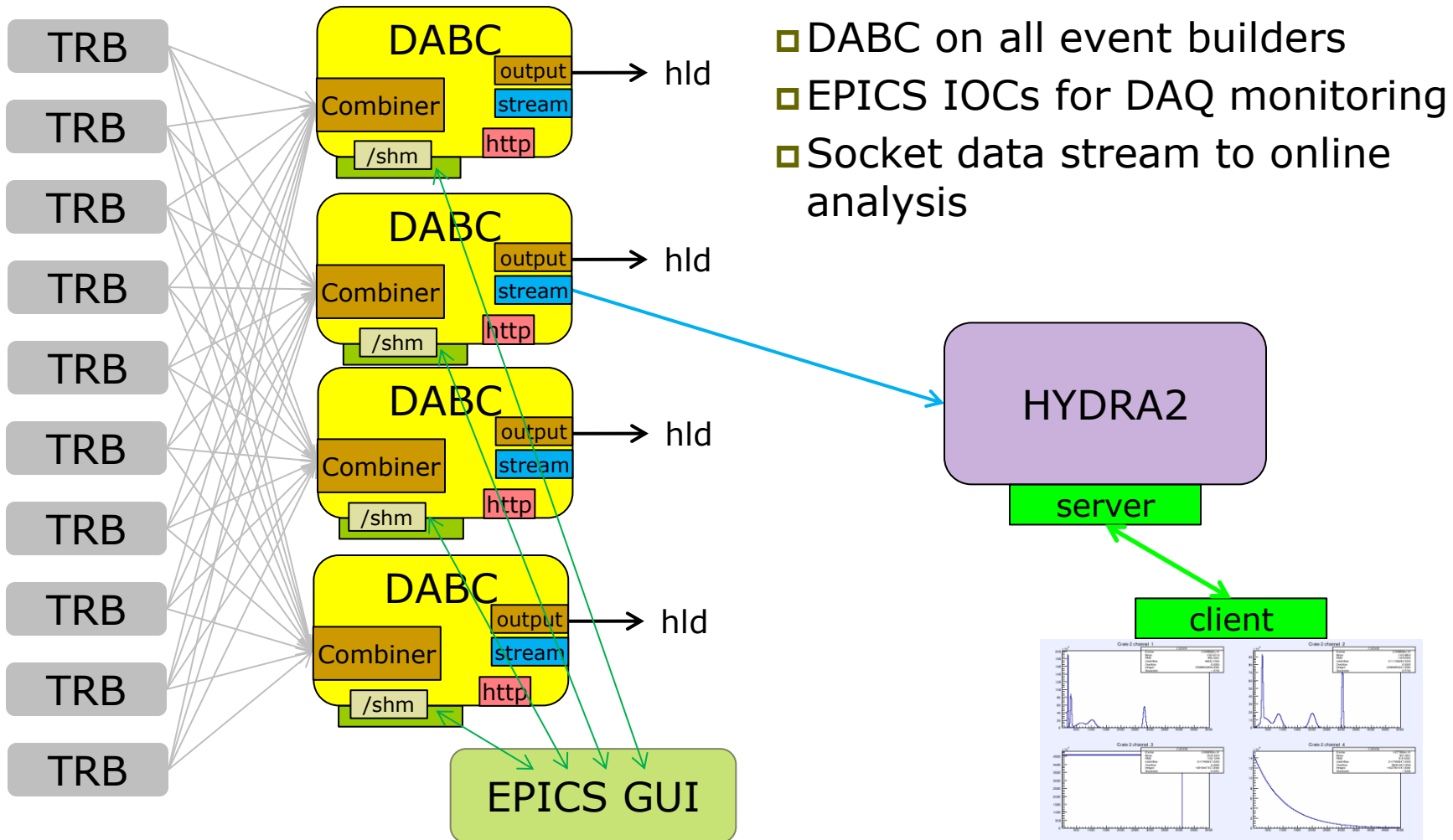
Outlook

- DABC in last beamtime
- DAQ upgrade possibilities
- THttpServer and JavaScript ROOT

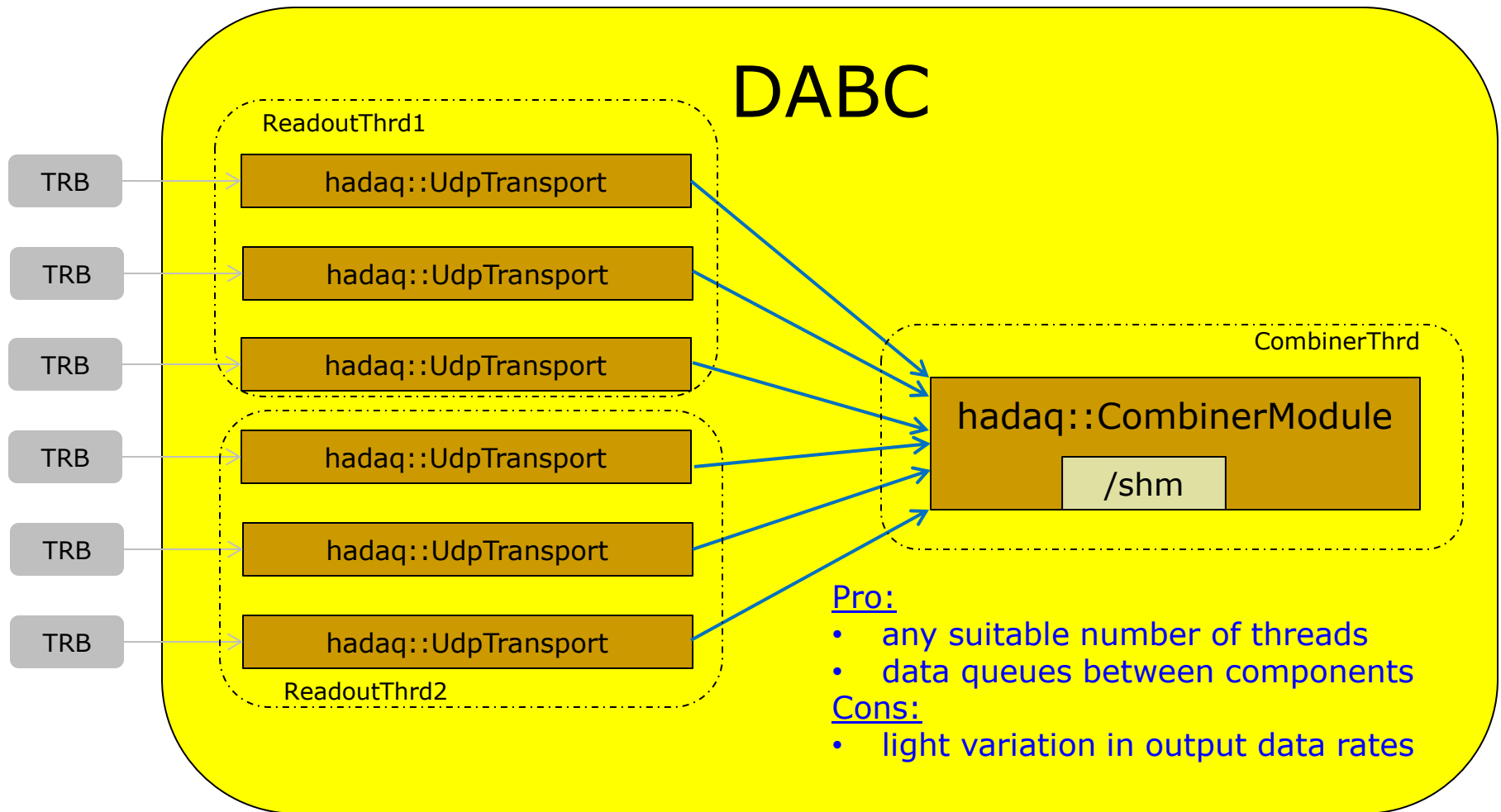
HADES DAQ before



HADES DAQ now



DABC event builder – many threads

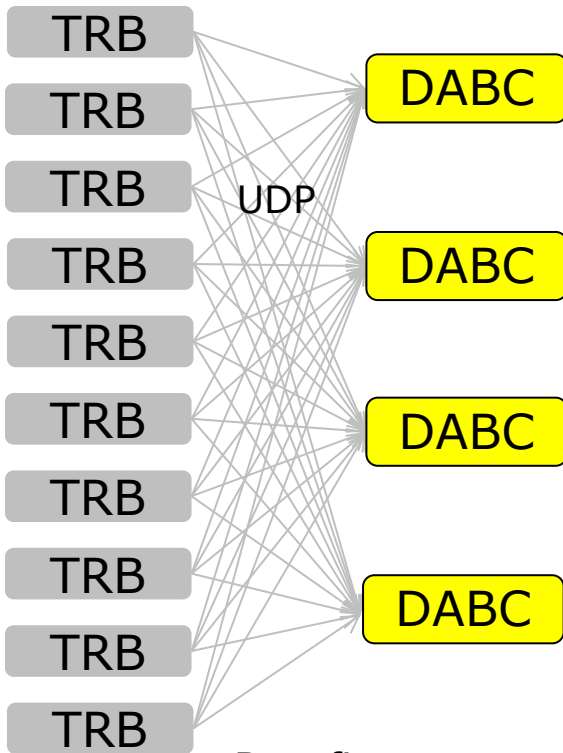


DAQ upgrade possibilities

- UDP-based event building
 - connection-less, therefore very flexible
 - potential data loss at high rates
 - event building can be done with DABC nodes

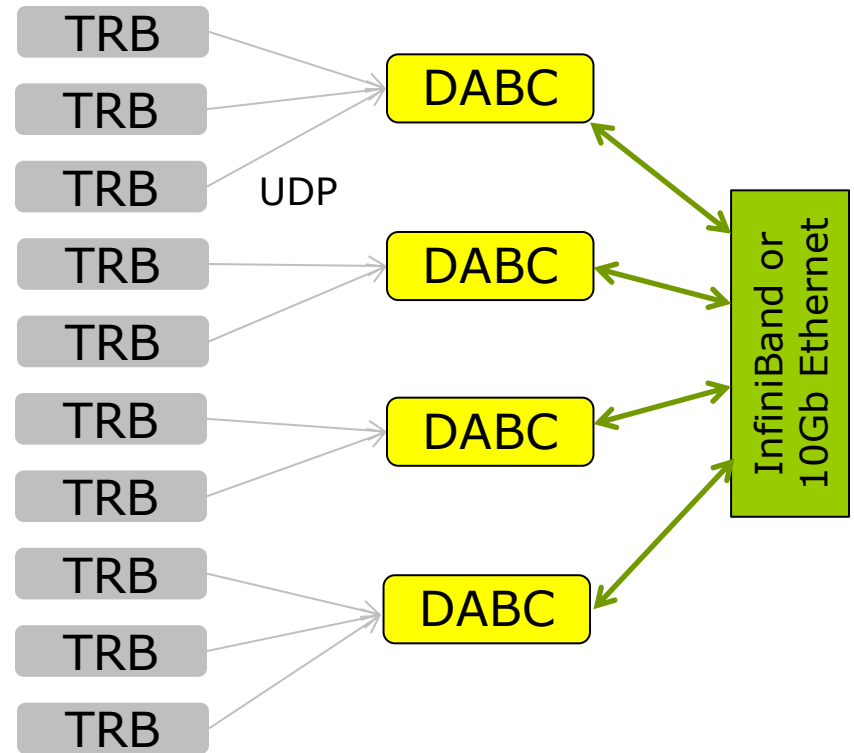
- TRB3/TDC calibrations
 - large volume in database ($\sim 2\text{KB}/\text{channel}/\text{day}$)
 - can be performed online
 - can be integrated in DABC event builder
 - requires changes in network topology

Change networking topology



Benefits:

- less probability of data lost on UDP
- potential possibility to exchange by lightweight TCP
- IB is much more powerful switching network
- possibility for online TDC calibrations



Cons:

- any DABC node could be single point of failure

THttpServer and JavaScript ROOT

- New development in ROOT
- Since June 2014 in official ROOT releases
- Provide generic web interface for arbitrary ROOT-based applications
- Very easy to build custom web pages with ROOT graphics inside
- Docs and many examples
 - <http://web-docs.gsi.de/~linev/js/>

THttpServer in ROOT

- ❑ http access to any ROOT-based application
- ❑ threads safe
- ❑ directly creates JSON representation
 - ROOT I/O remains on server side
- ❑ access to every data-member in registered objects
- ❑ JavaScript ROOT for visualization
- ❑ Demo page:
<http://web-docs.gsi.de/~linev/js/3.0/demo/>

http::Server in DABC

- ❑ Same approach as for ROOT
- ❑ Access to different data in DAQ application
- ❑ JavaScript ROOT as user interface
- ❑ One could integrate and access data from different sources:
 - DABC, ROOT, MBS, EPICS, DIM, FESA,...
- ❑ Same data could be accessed via DABC socket by Go4, one still able to use native ROOT graphics
- ❑ One also could use command-line tools to access DAQ/analysis (via socket or http)