Dyplo in Space

Mike Looijmans

Embedded in your future

Agenda

- Introduction

- What is Dyplo?

- Status of space-enabled Dyplo

  - Radiation hardened FPGA support

  - RTEMS support

- Questions

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

Mike Looijmans

▲ System Expert at TOPIC Embedded Products
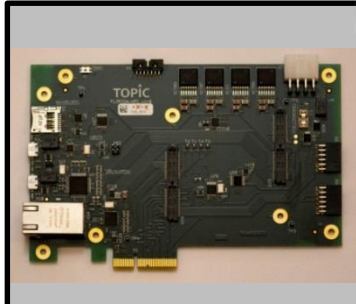
Topic today

- ▲ Real Embedded company; 170 employees
    - ▲ 125+ embedded software developers
    - ▲ 25+ FPGA designers
    - ▲ 10+ board designers
- ▲ Founded in 1996, privately owned
- ▲ 3 Business units:
    - ▲ Since 1996: Consultancy: the Netherlands
    - ▲ Since 2006: Project execution: Europe and North America
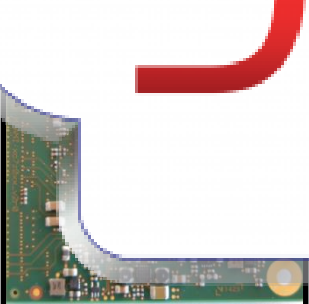   - ▲ Since 2014: Product development and sales: world wide

XILINX®

ALLIANCE PROGRAM
PREMIER MEMBER
1 of 10 worldwide

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

Topic Products - our ecosystem

FLORIDA
Powered by TOPIC

DYPLO®
Powered by TOPIC

DSIGN
Powered by TOPIC

Family of carrier boards	Family of system on modules	Dynamic Process Loader	Plug and Play IP-blocks	Fully tailor-made SW & HW solutions

Dynamic process loader
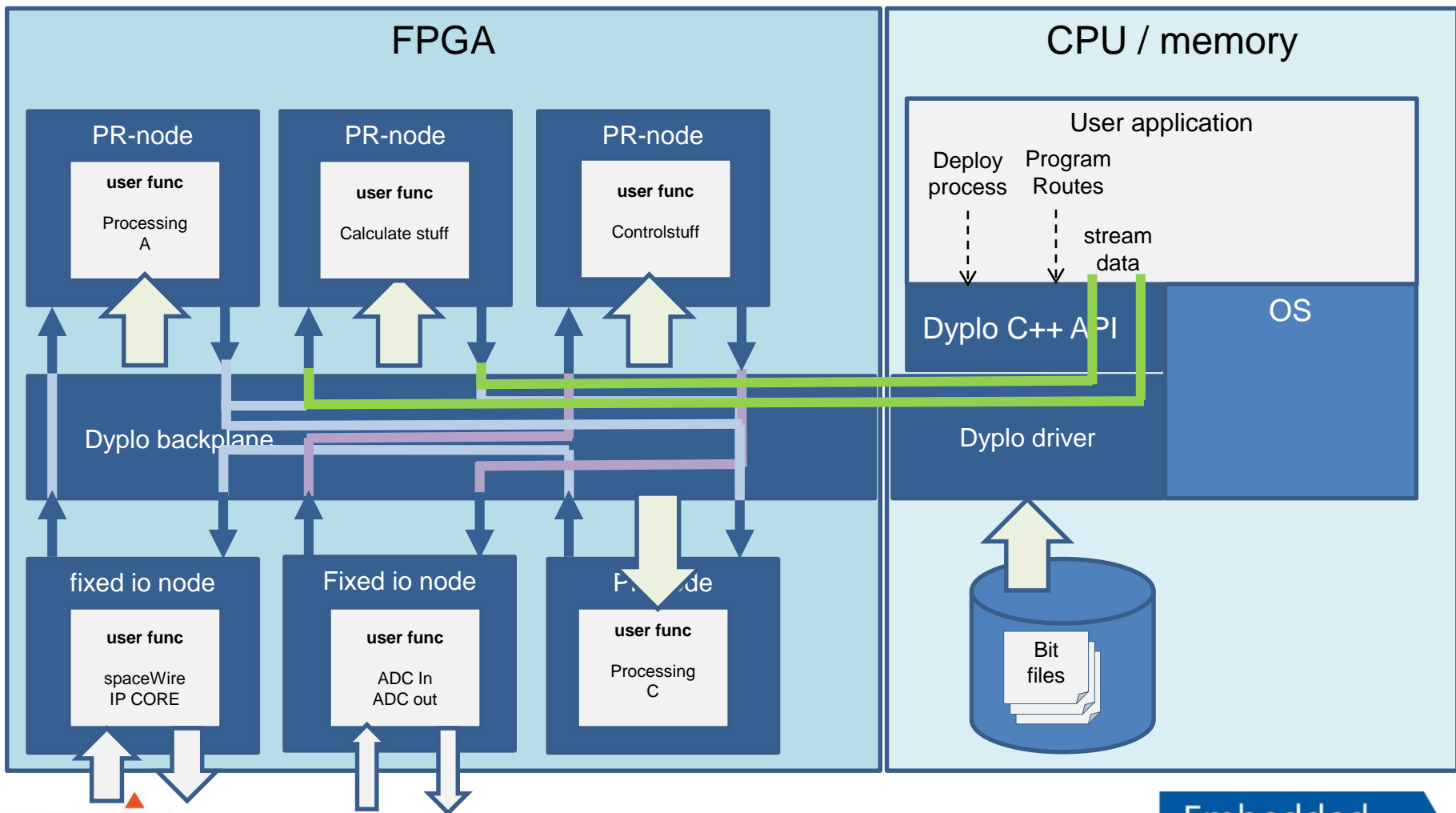
TOPIC
EMBEDDED SYSTEMS

Embedded
in your future

Dyplo provides …

- Dynamic loading of processes on the FPGA using partial reconfiguration
  - And static assignment
- Infrastructure for streaming data between these processes
  - With run-time routing
- Interfaces for streaming data between FPGA logic and CPU programs
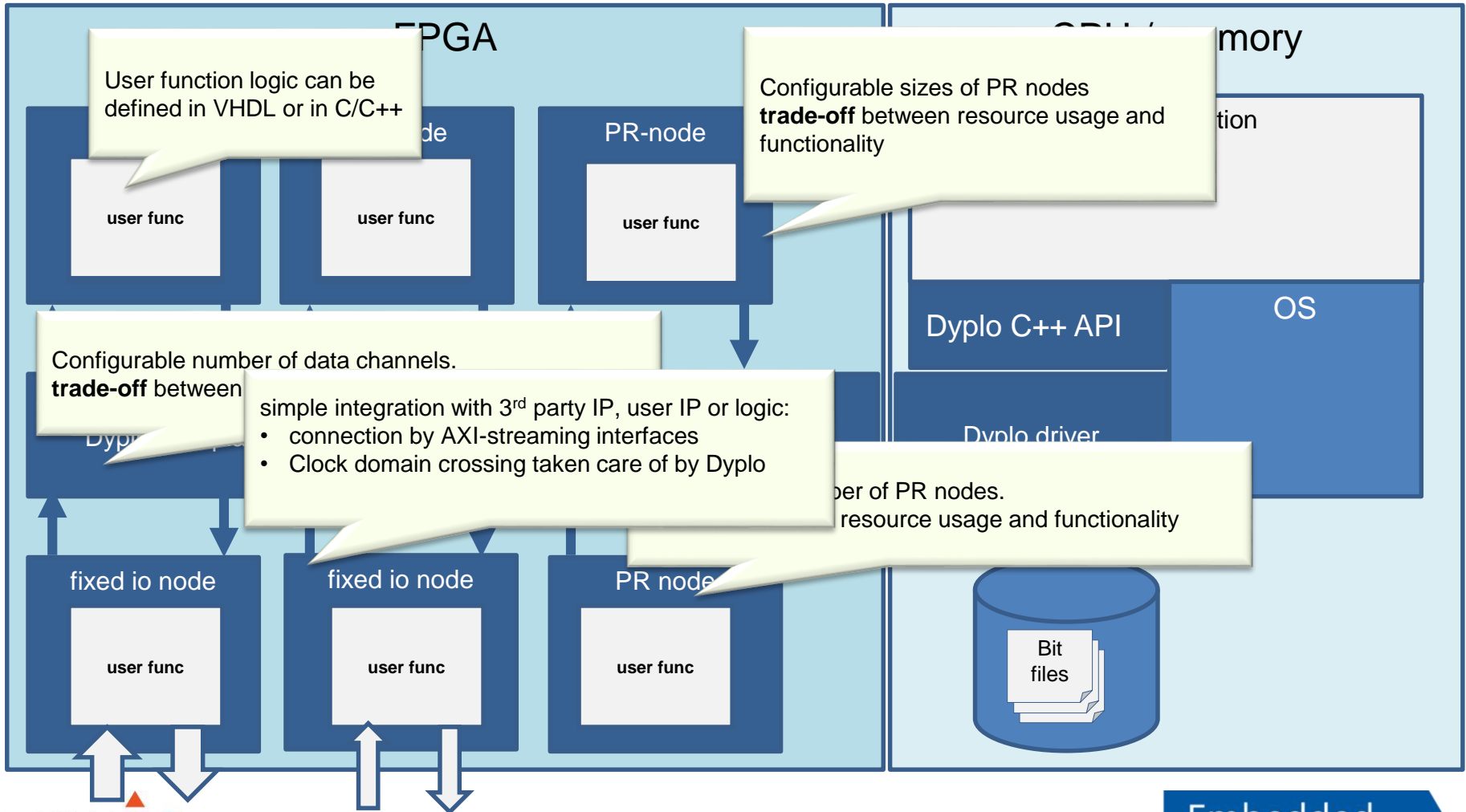 - latency, throughput, resource usage

- ▲ Use the FPGA as accelerator
  - ▲ Offload work to the FPGA
- ▲ Acquire data from external sources
  - ▲ Real-time
- ▲ Transfer data to external targets
  - ▲ Output and control

What is Dyplo

FPGA

PR-node
user func
Processing A

PR-node
user func
Calculate stuff

PR-node
user func
Controlstuff

Dyplo backplane

fixed io node
user func
spaceWire IP CORE

Fixed io node
user func
ADC In ADC out

PR-node
user func
Processing C

CPU / memory

User application
Deploy process
Program Routes
stream data

Dyplo C++ API

OS

Dyplo driver

Bit files

The Dyplo infrastructure itself is configurable

FPGA

CPU/memory

User function logic can be defined in VHDL or in C/C++

PR-node

PR-node

user func

user func

user func

Configurable sizes of PR nodes **trade-off** between resource usage and functionality

tion

Configurable number of data channels. **trade-off** between

Dyplo C++ API

OS

simple integration with 3rd party IP, user IP or logic:
- connection by AXI-streaming interfaces
- Clock domain crossing taken care of by Dyplo

Dyplo driver

ber of PR nodes.
resource usage and functionality

fixed io node

fixed io node

PR node

user func

user func

user func

Bit files

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

- ▲ Create Dyplo-enabled project
  - ▲ Or add Dyplo to existing project
- ▲ Configure Dyplo
  - ▲ Resources, streams, nodes, bandwidth
- ▲ Create "static" design
  - ▲ Connect I/O nodes to external components
- ▲ Add node functionality
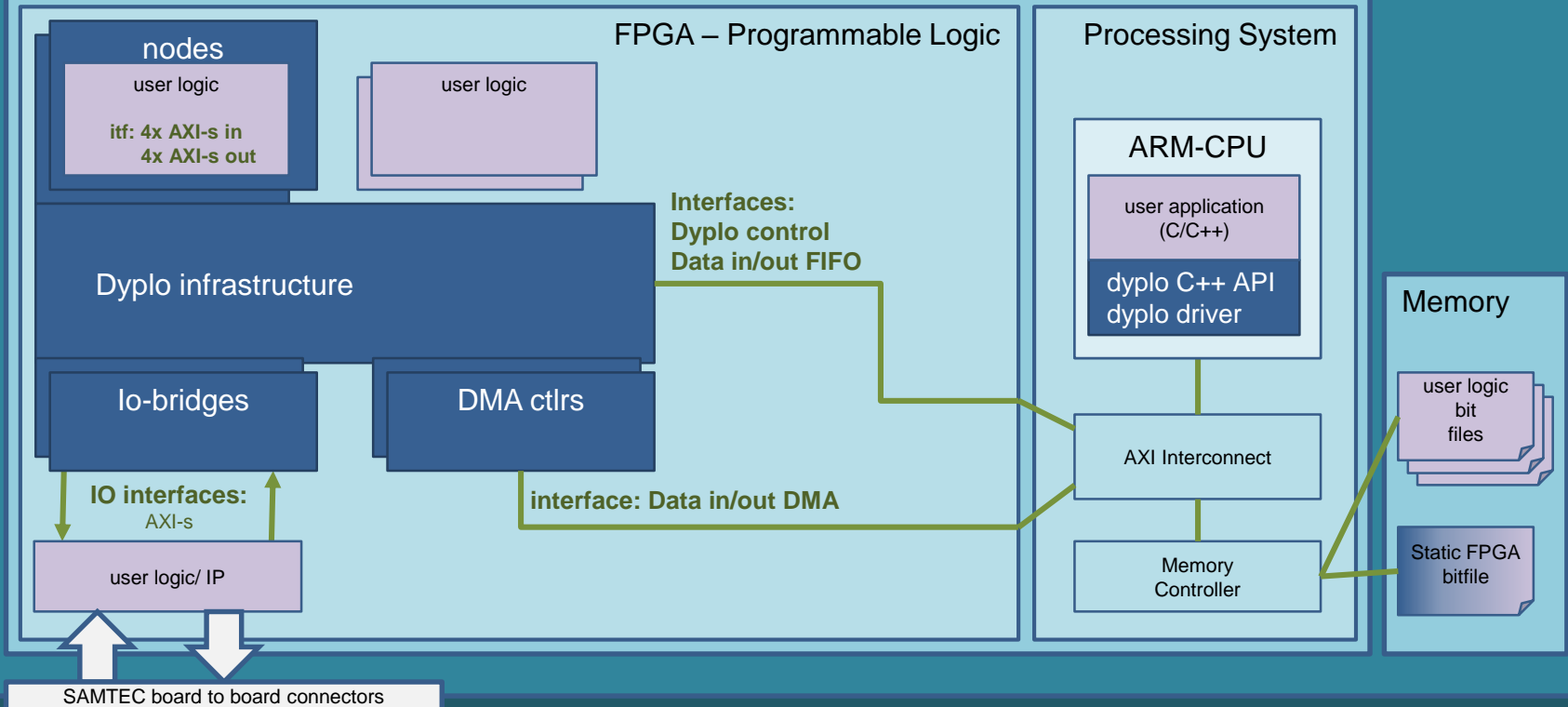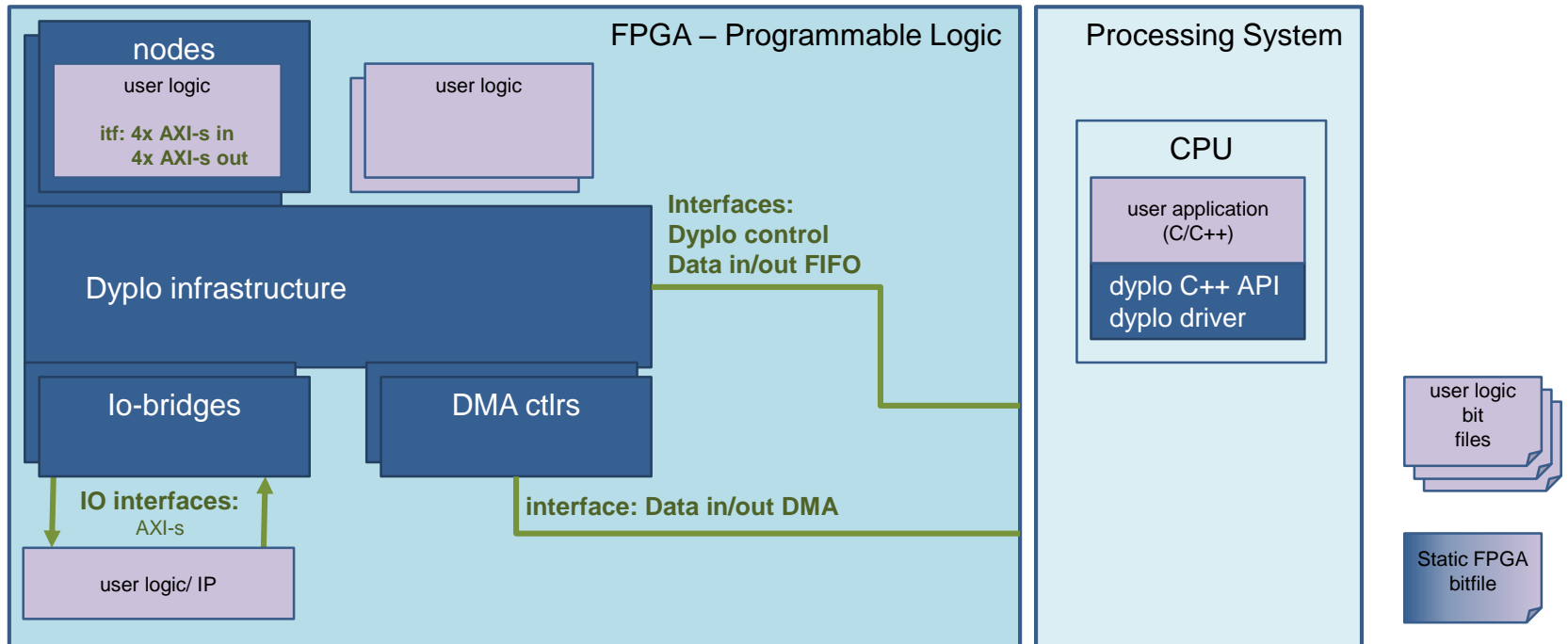 - ▲ C/C++ code or VHDL/Verilog
- ▲ More changes

TOPIC
EMBEDDED PRODUCTS
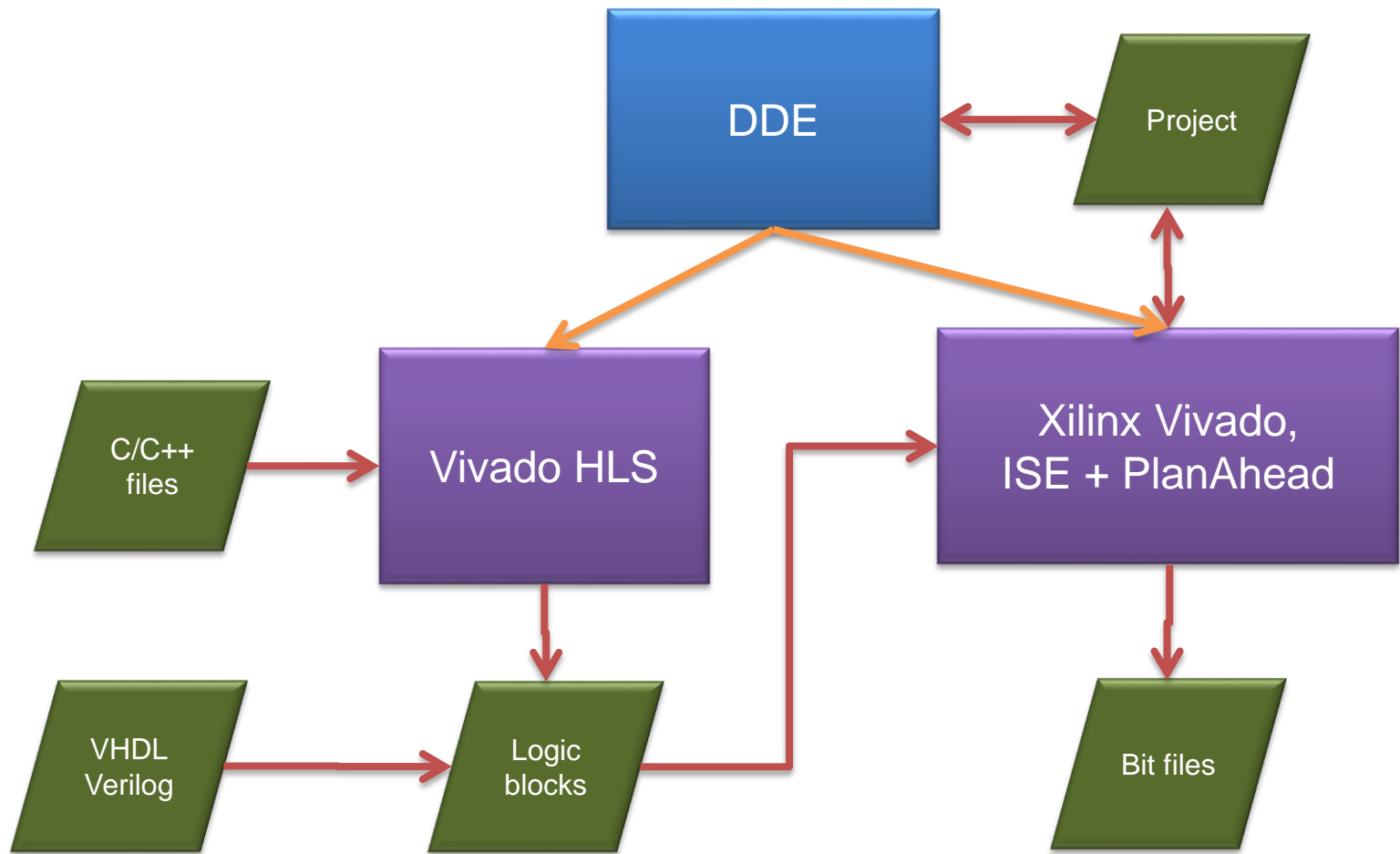
Embedded
in your future

Toolflow

Dyplo – on Zynq series

Dyplo – on Kintex series - 1

Dyplo – on Virtex 5 (ML506)

Xilinx Evalutation board: ML506

Xilinx XT50T

FPGA – Programmable Logic

nodes

user logic

**itf: 4x AXI-s in
4x AXI-s out**

user logic

Dyplo infrastructure

**Interfaces:
Dyplo control
Data in/out FIFO**

SOFTCORE
LEON2FT

user application
(C/C++)

dyplo C++ API
dyplo driver

Memory

user logic
bit
files

Io-bridges

DMA ctlrs

Static FPGA
bitfile

IO interfaces:
AXI-s

user logic/ IP

interface: Data in/out DMA

SpaceWire Adapter

Dyplo driver

- Available for Linux, Windows and RTEMS

- Driver represents Dyplo infrastructure in posix file interface

- Directly usable from command shell
  - Example: stream content from file directly to a datastream on the backplane: `cat file > /dev/dyplow0`

- C++ API available for applications
  - Other language APIs can be made (using ioctl)

- Open Source (GITHUB)
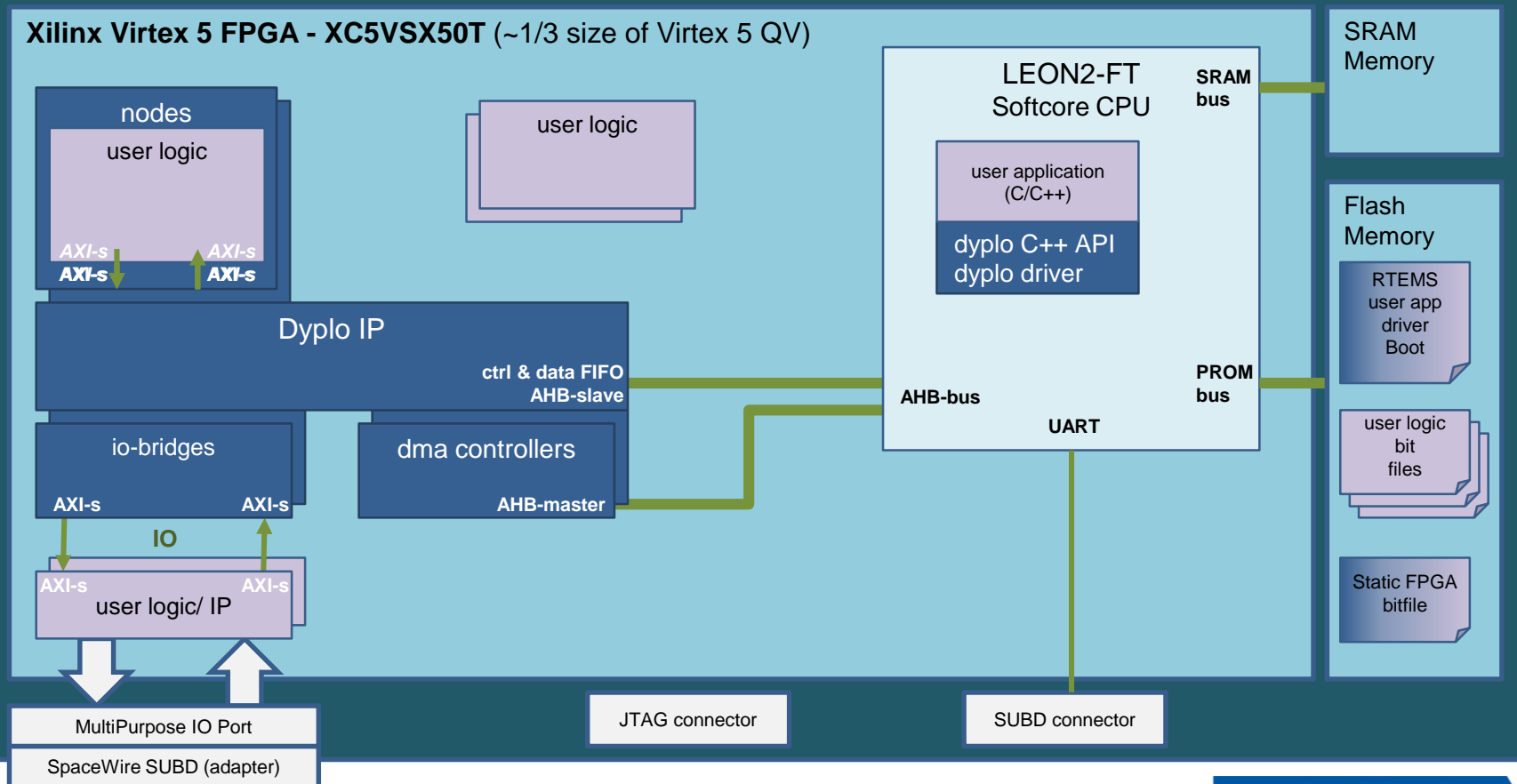
- Make Dyplo IP compatible with radiation hardened Xilinx Virtex 5QV

- Add RTEMS support

- Demonstrate
  - Running demo
  - LEON softcore
  - Image processing
  - SpaceWire

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

Dyplo – on Virtex 5 (ML506)

Xilinx Evaluation board: ML506

Xilinx Virtex 5 FPGA - XC5VSX50T (~1/3 size of Virtex 5 QV)

nodes
user logic

AXI-s   *AXI-s*
AXI-s   **AXI-s**

Dyplo IP

ctrl & data FIFO
AHB-slave

io-bridges

AXI-s   AXI-s

IO

AXI-s   AXI-s
user logic/ IP

dma controllers

AHB-master

user logic

LEON2-FT
Softcore CPU

user application
(C/C++)

dyplo C++ API
dyplo driver

SRAM bus

AHB-bus

PROM bus

UART

SRAM Memory

Flash Memory

RTEMS
user app
driver
Boot

user logic
bit
files

Static FPGA
bitfile

MultiPurpose IO Port

SpaceWire SUBD (adapter)

JTAG connector

SUBD connector

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

Thus far…

- Dyplo driver (and C++ lib) for RTEMS

- Partial reconfiguration on Virtex 5 with Dyplo concept

- But: ISE13.2 version required for Virtex5 QV
  - Xilinx has stopped support
  - Immature w.r.t. partial reconfiguration
  - Not all Dyplo functionality works
    - Difference windows/linux
    - Combining functionality in fixed and partial nodes

TOPIC
EMBEDDED PRODUCTS

Embedded
in your future

- Creating a SOC with LEON2-FT uses >15% of logic

- Overhead wrapper

  - (400 LUTS/1BlockRAM) relatively high on our smaller test breadboard (1/3 logic of QV)

Dyplo roadmap

- Working with ESA on Dyplo
  - Dyplo driver for RTEMS available medium 2017
  - Dyplo support for Virtex 5 QV (Radiation hardened) available medium 2017
- Ideas for Dyplo FT (Fault Tolerant)

Dyplo performance

- Dyplo infrastructure
    - Transports data between nodes
    - "Backplane"

- Node <-> Dyplo backplane bandwidth
    - Data from backplane to node and vice versa

- CPU <-> Dyplo backplane bandwidth
    - From CPU/DMA node to CPU (or memory)

- Dyplo infrastructure bandwidth

- Mostly determined by clock

  - 100 MHz on low end fabric (e.g. Artix)

  - Can be over 200MHz (e.g. Kintex)

One 32-bit word per clock per lane

- 1 .. 4 lanes

- Lowest 400 Mbyte/s   (3,2 Gbit/s)   for Zynq 7xxx at 100 Mhz
- Highest 1600 Mbyte/s   (12,8 Gbit/s)   for Zynq 7xxx at 100 Mhz

Dyplo performance - node

▲ Node <-> Dyplo backplane bandwidth

▲ Clock speed

  ▲ Same as backplane clock

▲ All streams share the bandwidth

  ▲ 400 Mbyte/s   (3.2 Gbit/s)   e.g. Zynq 7015 at 100 Mhz
  ▲ 800 Mbyte/s   (6.4 Gbit/s)   e.g. Zynq 7030 at 200 Mhz

▲ Input and output simultaneous

Dyplo performance - CPU

▲ CPU <-> Dyplo backplane bandwidth

▲ Using memory mapped FIFO

  ▲ On Zynq7xxx: 25 MB/s (0.2Gbps)

  ▲ On PCIe card: 3 MB/s read, 20MB/s write (due to bus latency)

▲ Using DMA

  ▲ On Zynq7xxx: 600 MB/s (limited by HP port)

    ▲ 1200MB/s through ACP when targeting L2 or L1 cache

  ▲ On PCIe: 800MB/s (limited by design)

    ▲ Theoretical: 2.5 GB/s for 4-lane gen2

Dyplo performance - resources

▲ FPGA resource usage depends on user configuration

▲ CPU interface

  ▲ ~ 4000 registers   *(157200 available @ XC7Z030 =<2.6%)*

  ▲ ~ 2500 LUT6   *(78600 available @ XC7Z030 =<3.1%)*

  ▲ 16 BRAM36   *(265 available   @ XC7Z030 =<6.3%)*

▲ node wrappers

  ▲ ~ 75-300 registers   per node (*=<0.2%*)

  ▲ ~ 50-200 LUT6   per node (*=<0.3%*)

  ▲ 1 BRAM36 per input stream   per node (*=<0.4%*)

Questions

?