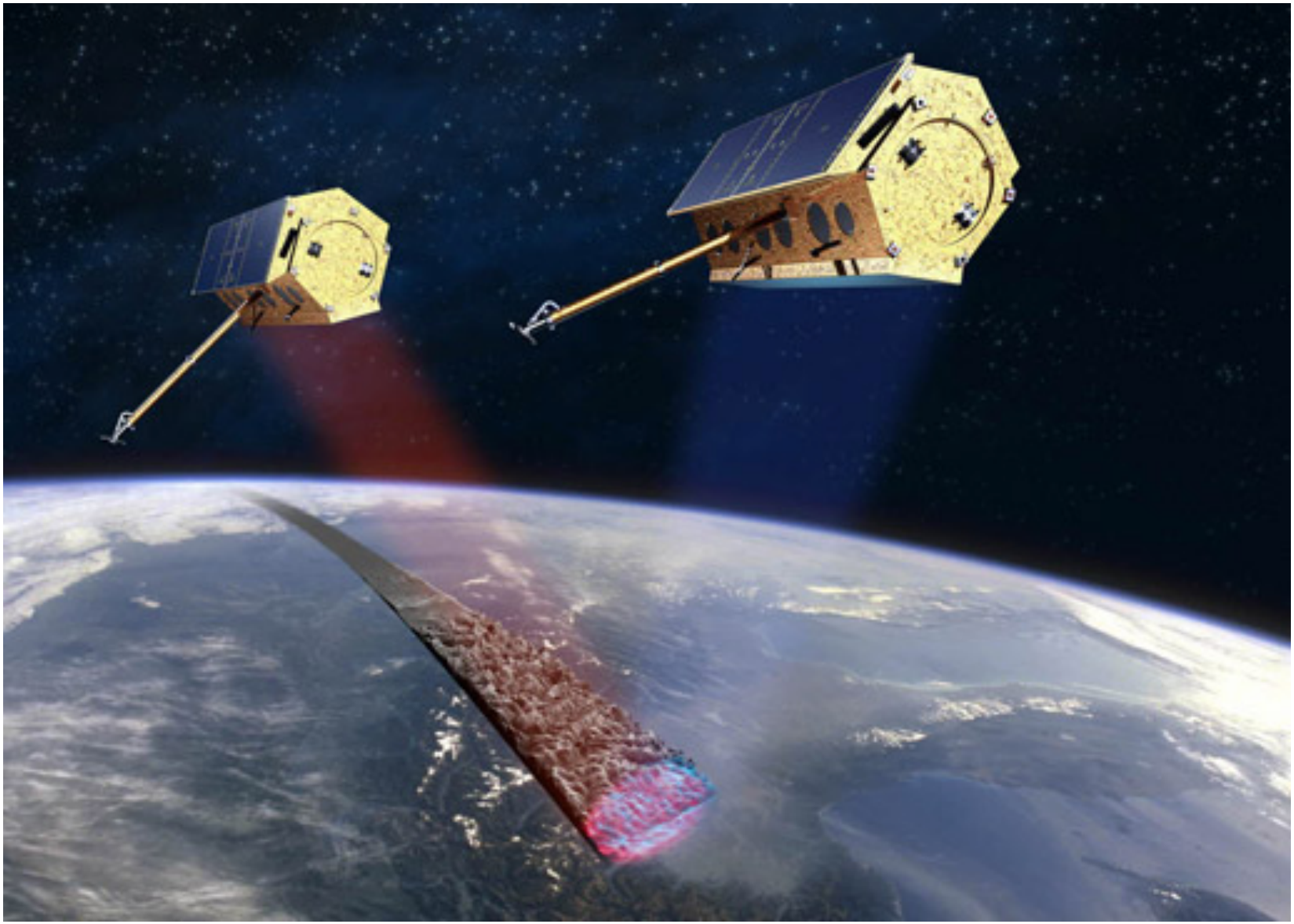
TU DELFT SPACE INSTITUTE

Increasing computing performance of ADCS subsystems in small satellites for earth observation

Johan Carvajal-Godínez, Morteza Haghayegh, Allan Granados, Jaan Viru and Jian Guo
Space Engineering Department
Faculty of Aerospace Engineering
Delft University of Technology

Outline

- Introduction
- ADCS challenges
- “Y model” approach for ADCS
- DelFFi ADCS modeling and simulation
- OBC Hardware selection
- ADCS software integration
- Conclusions



What does Earth Observation and Formation Flying have in common?

Challenging ADCS subsystem

- ✓ Multiple attitude modes
- ✓ High pointing accuracy
- ✓ Precise three axis control algorithms
- ✓ High resolution data types from sensors
- ✓ Onboard sensor calibration
- ✓ Fault detection and correction
- ✓ onboard functions for autonomous operation

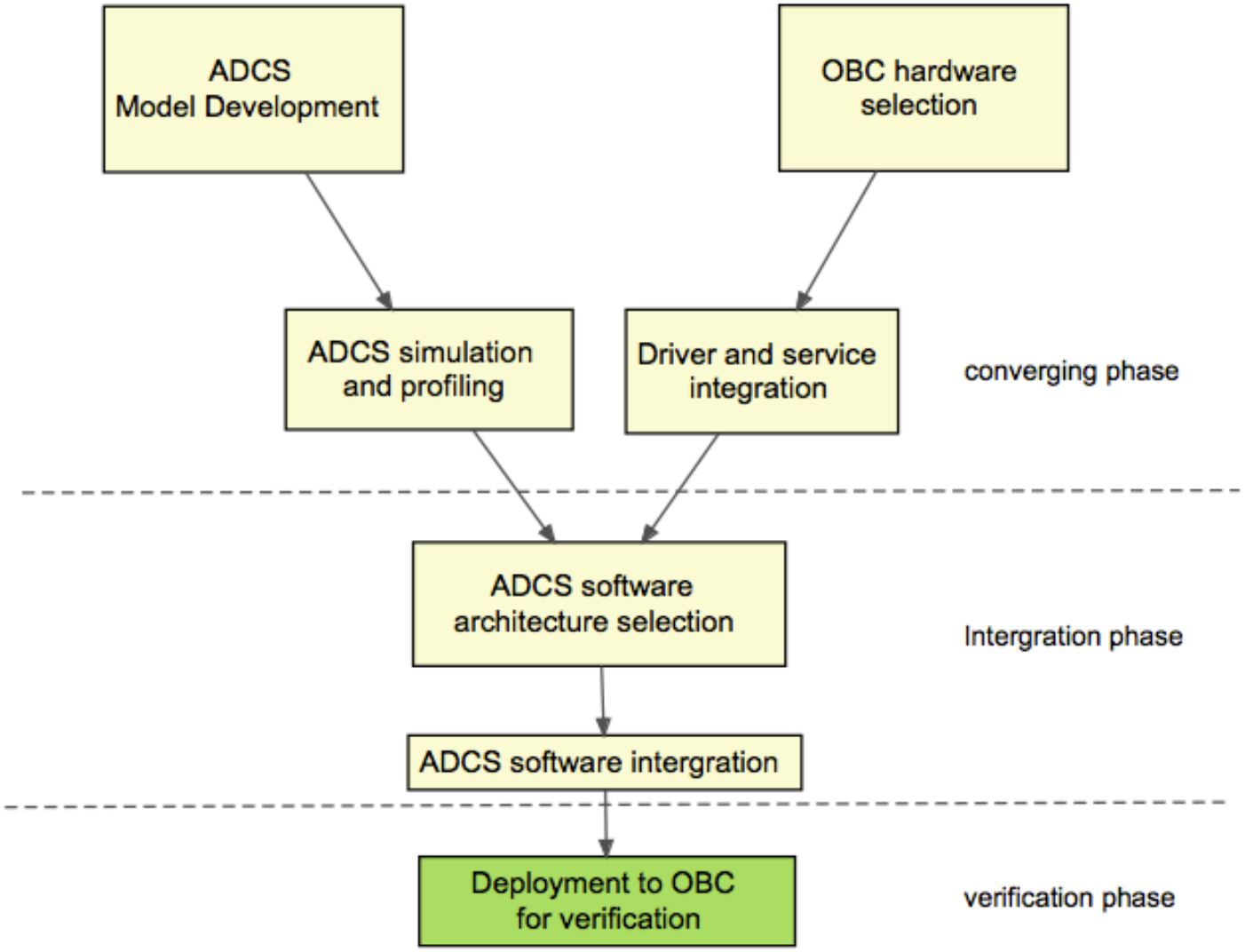
More complex software → **improving OBC capabilities for ADCS**

DelFFi Mission Statement

"The DelFFi mission shall demonstrate **autonomous formation flying** and provide enhanced scientific return within QB50 from 2016 onwards, by utilizing **two identical triple-unit Cubesats** of TU Delft which further advance the Delfi-n3Xt platform."

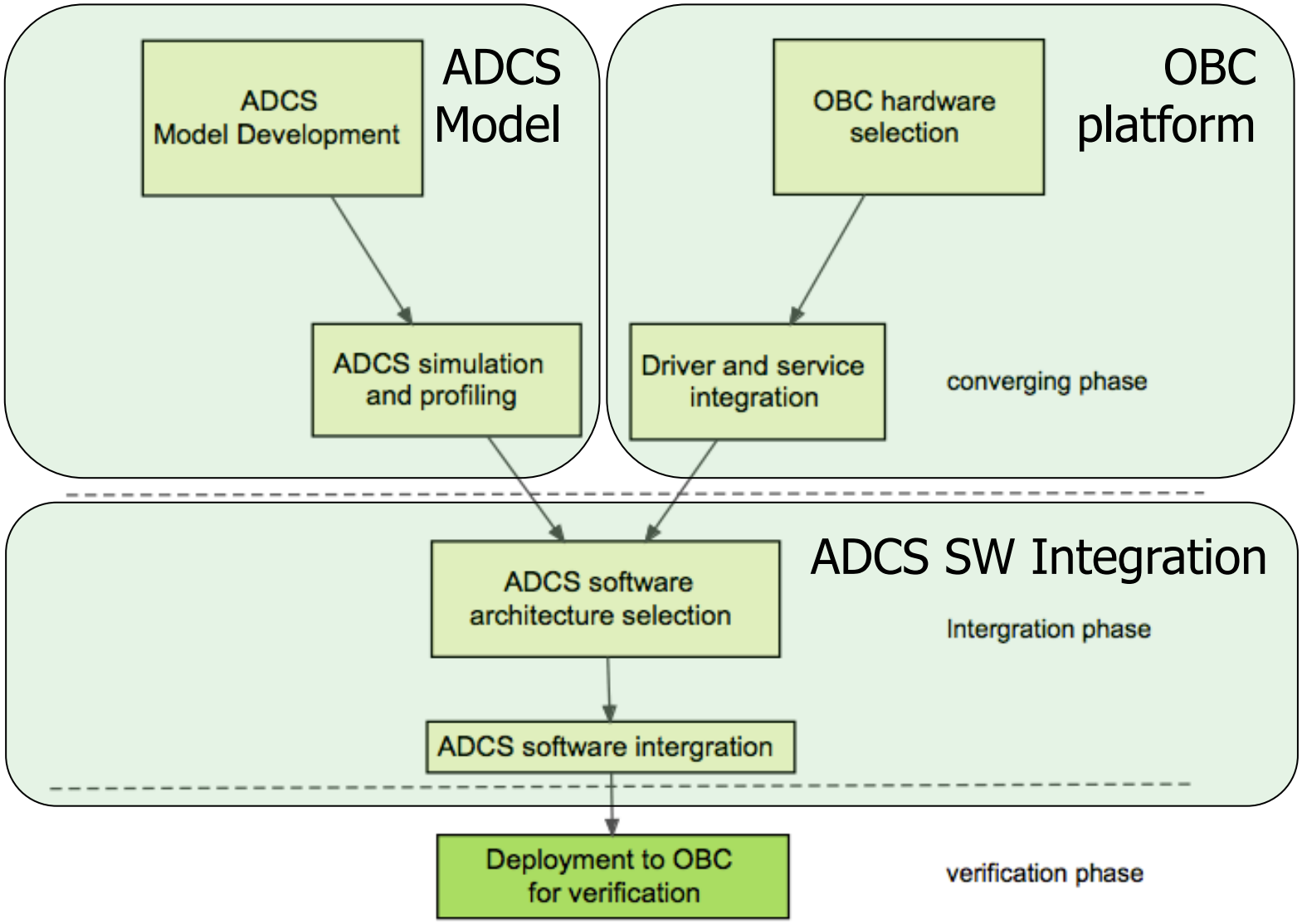
Source: SSE TU Delft

DelFFi ADCS software development

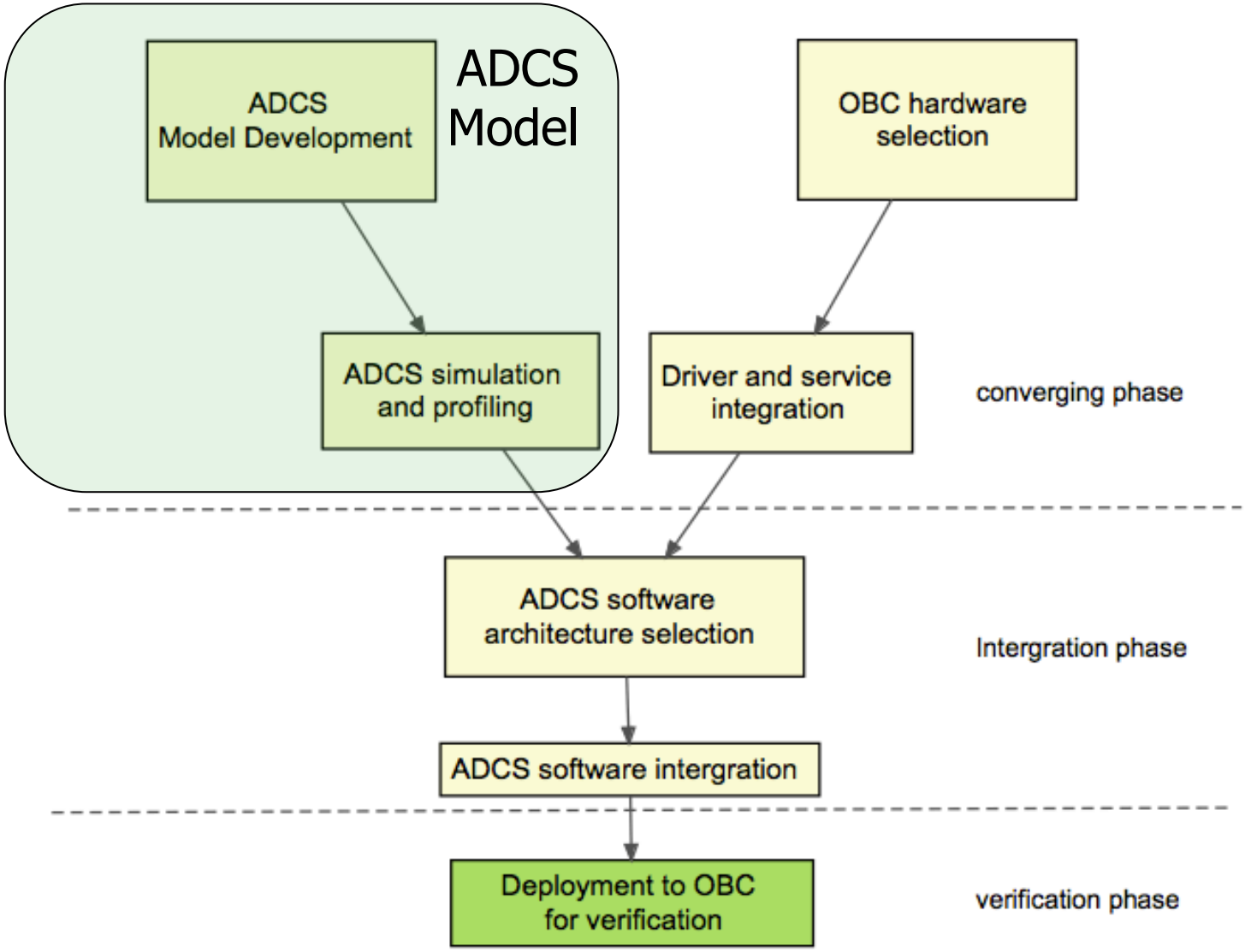


"Y approach"

DelFFi ADCS software development



DelFFi ADCS simulation model

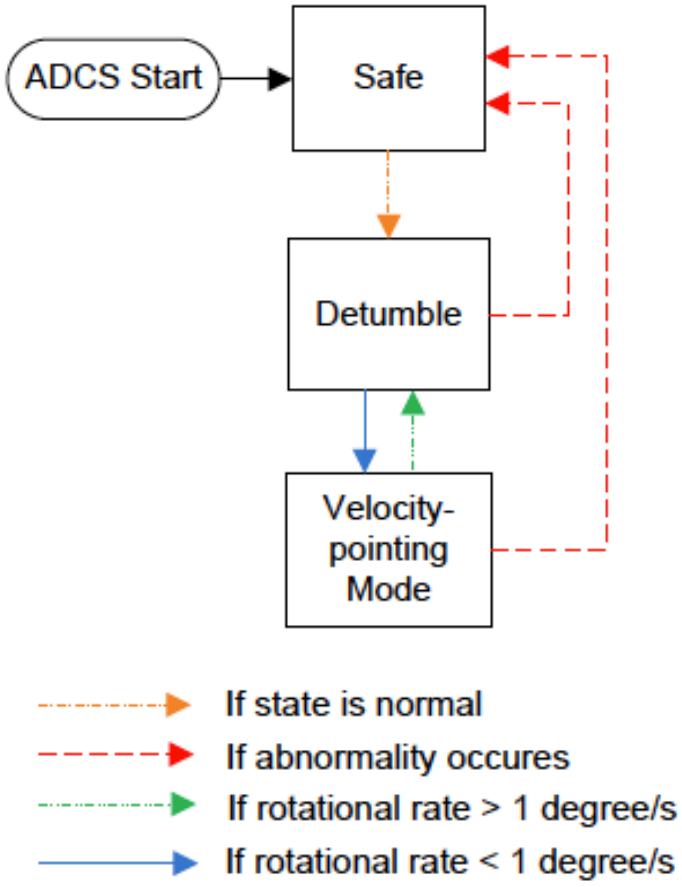


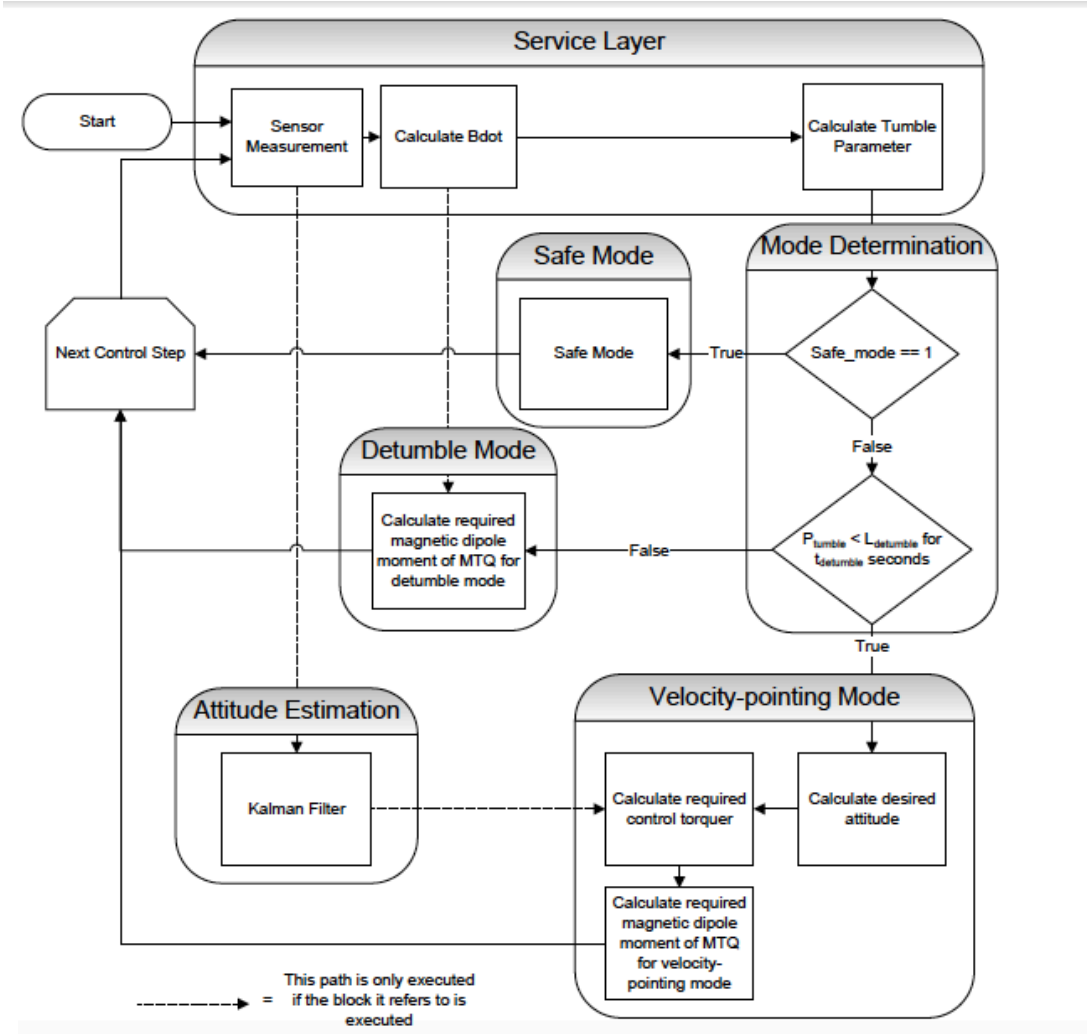
DelFFi ADCS input requirements

ADCS Modes	Threshold	Control Objective	Sensor	Actuator
Detumbling	>1°/s	1°/s	Magnetometer	Magnetorquer
Nadir pointing	<1°/s	<1°/s; <10°	Magnetometer, Sun Sensors	Magnetorquer
Sun Pointing	Command	<1°/s; <10°	Magnetometer, Sun Sensors	Magnetorquer
Thrust Vector Control (VP)	Command	<1°/s; <2°	Magnetometer, Sun Sensors	Magnetorquer, Reaction Wheel
Manuevering	Command	None	Magnetometer	Magnetorquer, Reaction Wheel
Safe	Command	None	Magnetometer	None

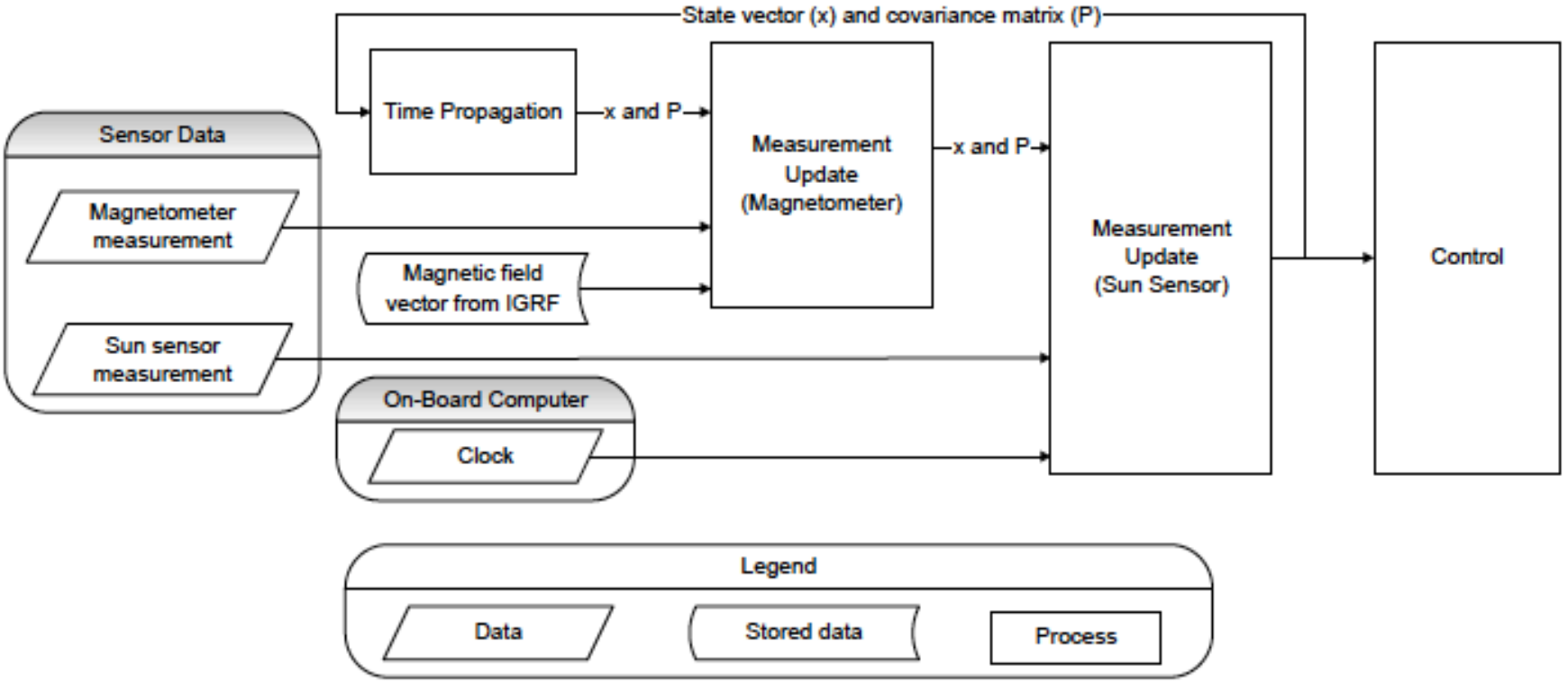
VP: Velocity pointing

ADCS software architecture (initial)





DelFFi ADCS simulation model



ADCS model profiling

Goal: Identifying the most demanding blocks inside the ADCS model by measuring relative CPU time utilization during simulation, for later code acceleration with digital signal processor (DSP)

Process Steps:

- Implement ADCS simulation model in Simulink
- Setup up the Matlab profiler to collect model performance data
- Setup up the simulation environment for ADCS model
- Run the model profiler
- Analyze ADCS model performance data
- Select most demanding model block for code acceleration with DSP

Simulation environment setup

Parameter	Value	Unit
Initial quaternion	$\begin{bmatrix} 0.378 & -0.378 & 0.756 & 0.378 \end{bmatrix}^T$	[-]
Orbit type	Circular	
Eccentricity	0	[-]
Inclination	98.6	degree
Right ascension of the ascending node	-15	degree
Argument of perigee	0	degree
Initial mean anomaly	0	degree
Satellite inertia	$\begin{bmatrix} 0.017 & 0 & 0 \\ 0 & 0.055 & 0 \\ 0 & 0 & 0.055 \end{bmatrix}$	$kg \cdot m^2$
Satellite dimensions	$\begin{bmatrix} 0.1 & 0.34 & 0.1 \end{bmatrix}^T$	m
Center of pressure offset from center of gravity	$\begin{bmatrix} 0.005 & 0.005 & 0.005 \end{bmatrix}^T$	m
Drag coefficient	2.2	[-]
Maximum magnetic torquer dipole moment	$\begin{bmatrix} 0.4 & 0.4 & 0.4 \end{bmatrix}^T$	$A \cdot m^2$
Residual magnetic dipole moment	$\begin{bmatrix} 0.005 & 0.005 & 0.005 \end{bmatrix}^T$	$A \cdot m^2$
Magnetometer bias	500	nT
Magnetometer noise standard deviation	170	nT
Sun sensor bias	< 3	degree
Sun sensor noise standard deviation	0.4	degree

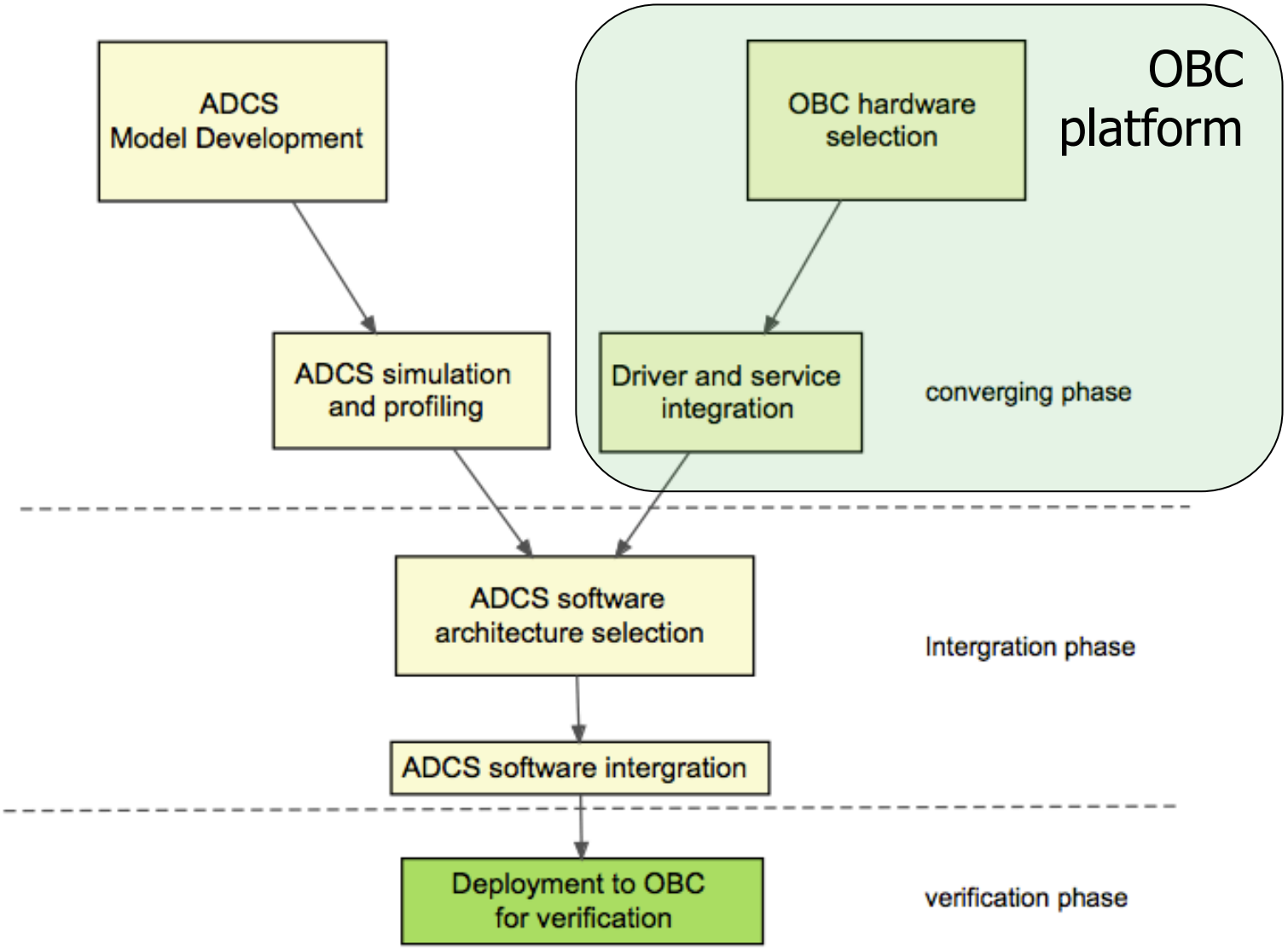
ADCS model profiling results

Block Function	Number of calls	Percentage of relative CPU time usage during simulation
Attitude estimation (EKF)	110 000	**52%**
Velocity Pointing mode	110 000	14%
Other OBC functions	110 000	34%

Other functions include:

- Environment initialization
- ADCS Mode determination algorithm
- Other attitude modes (Safe and De-Tumbling)
- Simulation Termination

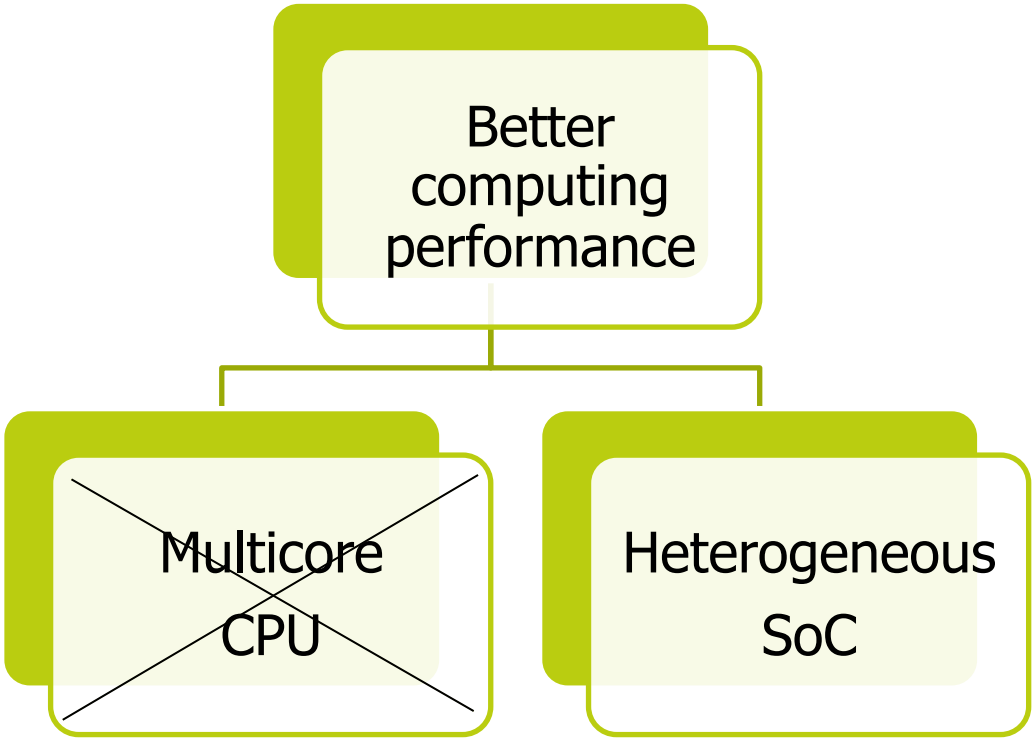
DelFFi OBC selection



OBC architecture trade off

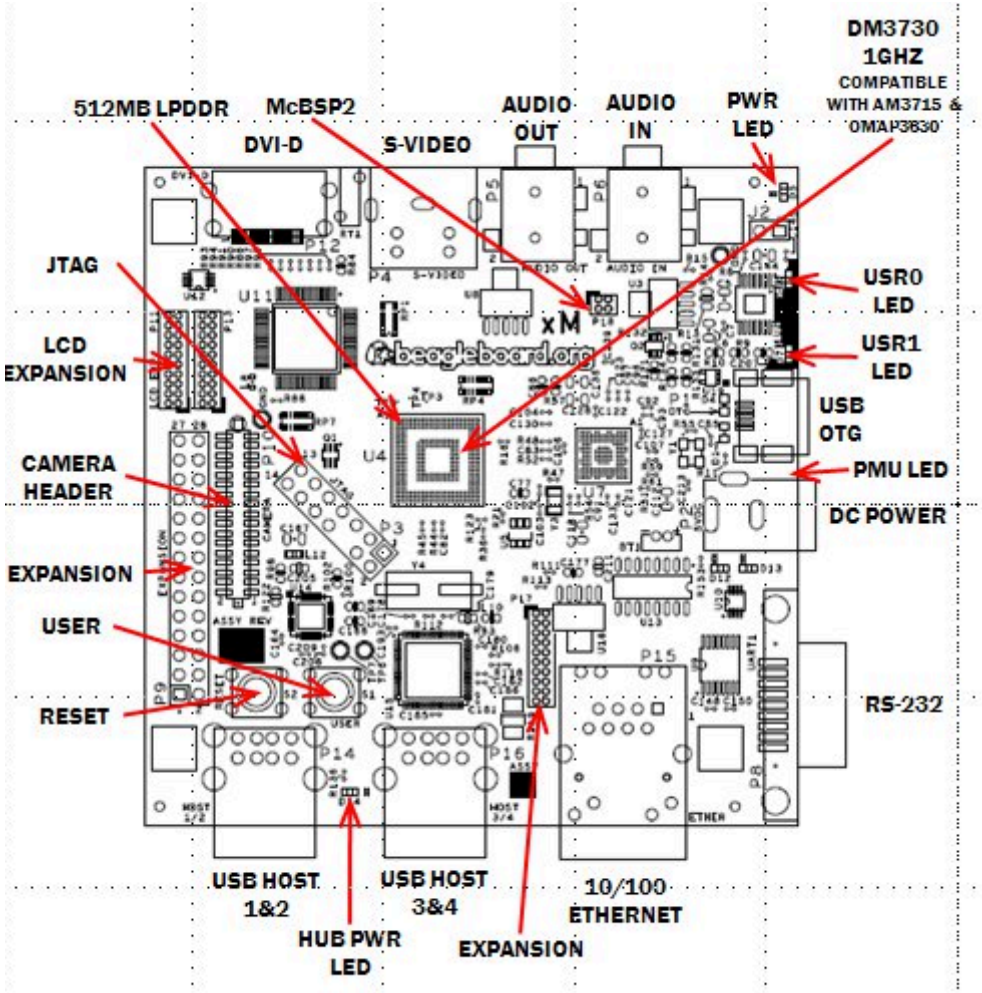
OBC Requirements:

- Code acceleration support
- Build system support from open embedded community
- I2C, SPI and UART support
- Power efficient floating point unit performance w.r.t. FPGA
- COTS available
- Open hardware and software



Beagleboard XM

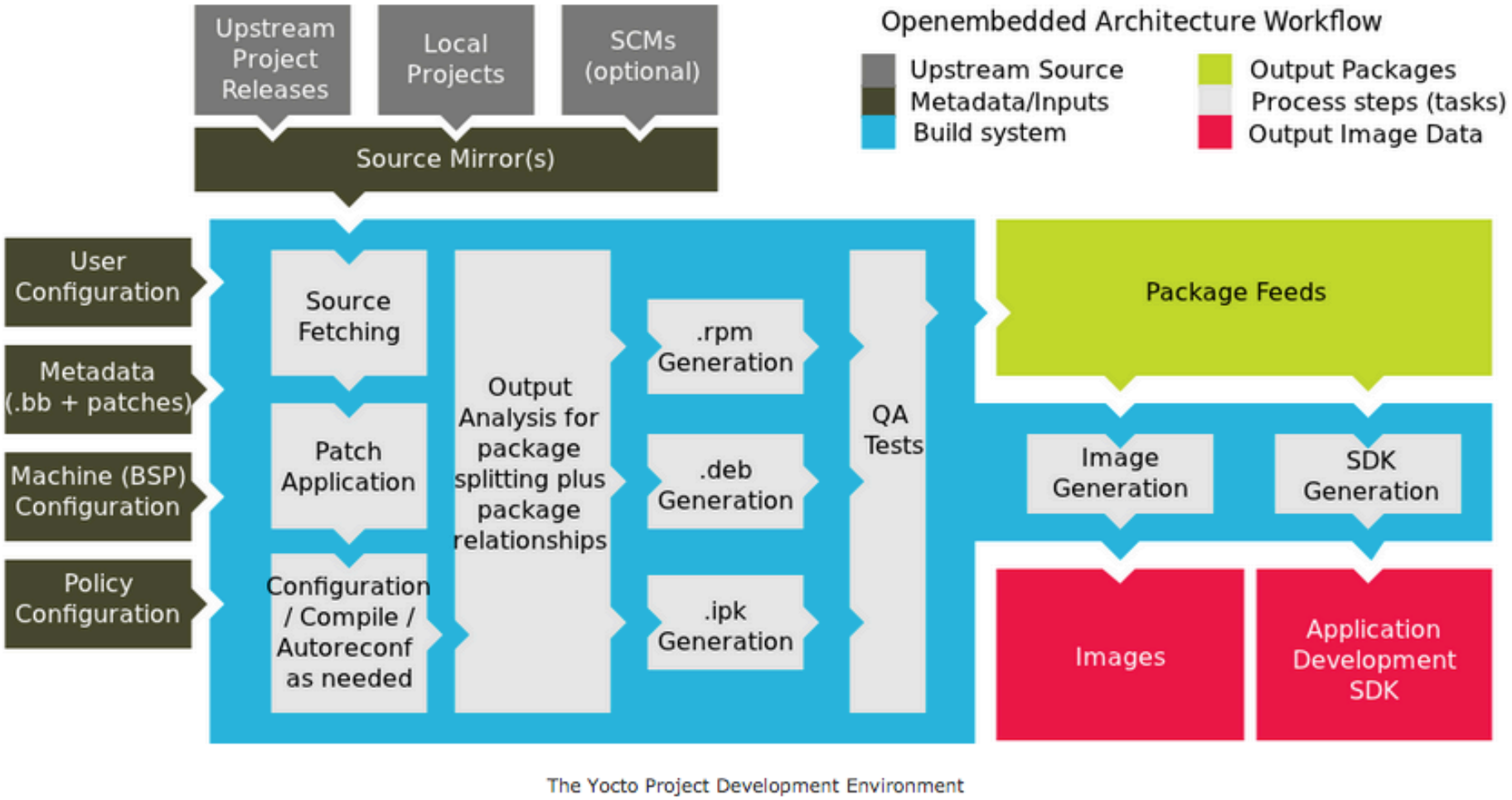
Hardware Test bed



Beagleboard XM:

- COTS and open hardware/software platform
- TI DM3730 SoC **(ARM processor +Digital Signal Processor)**
- Build system support from embedded community **(Yocto project)**
- Continuous integration support **(Jenkins)**

ADCS software build system



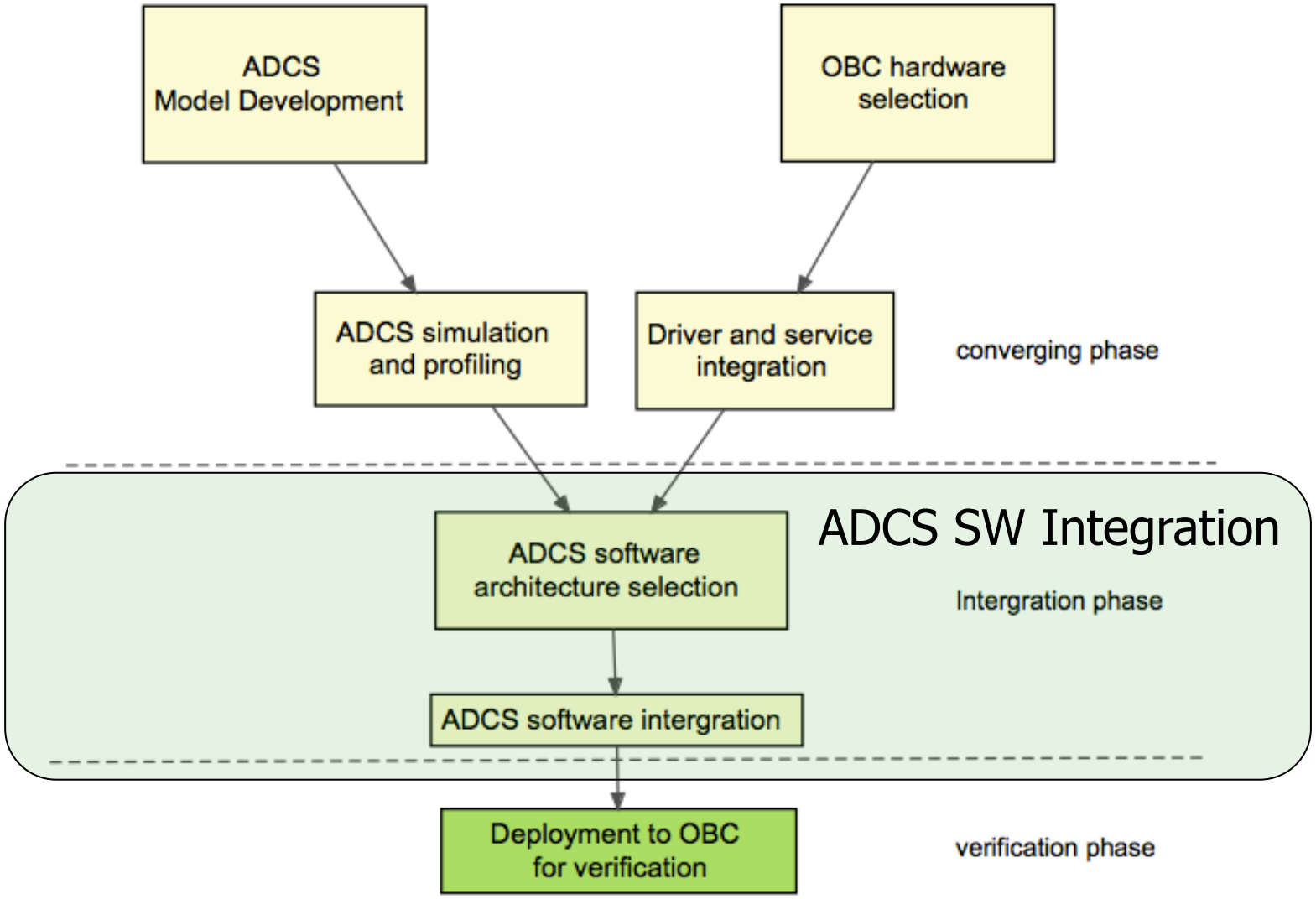
The Yocto Project Development Environment

Results on services and drivers support

Image	Size
MLO	48K
u-boot.img	369K
u-boot-spl.bin	47K
uImage	4,1M
uImage-omap3-beagle-xm.dtb	57K
FileSystem	13M
Total	18M

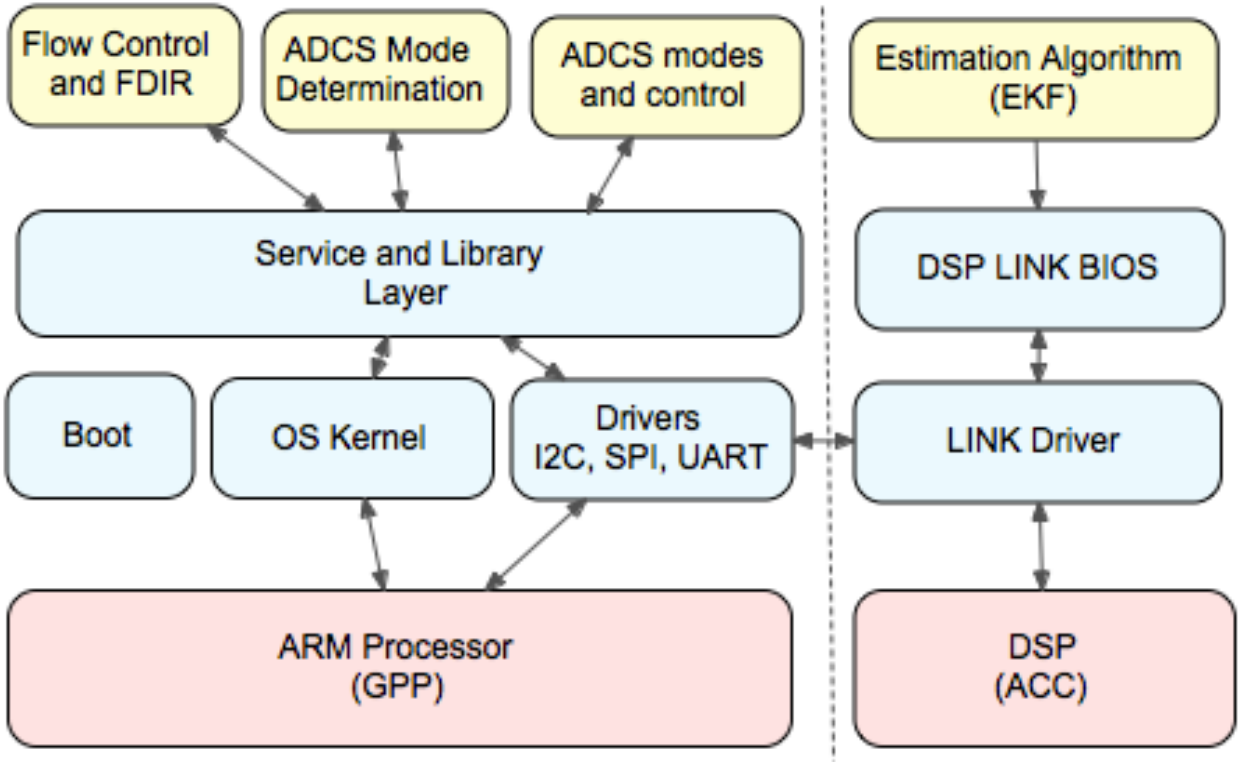
Operating system size with all driver support is 18 MB

DelFFi ADCS software development



DelFFi ADCS architecture (final)

Non Accelerated blocks Accelerated block

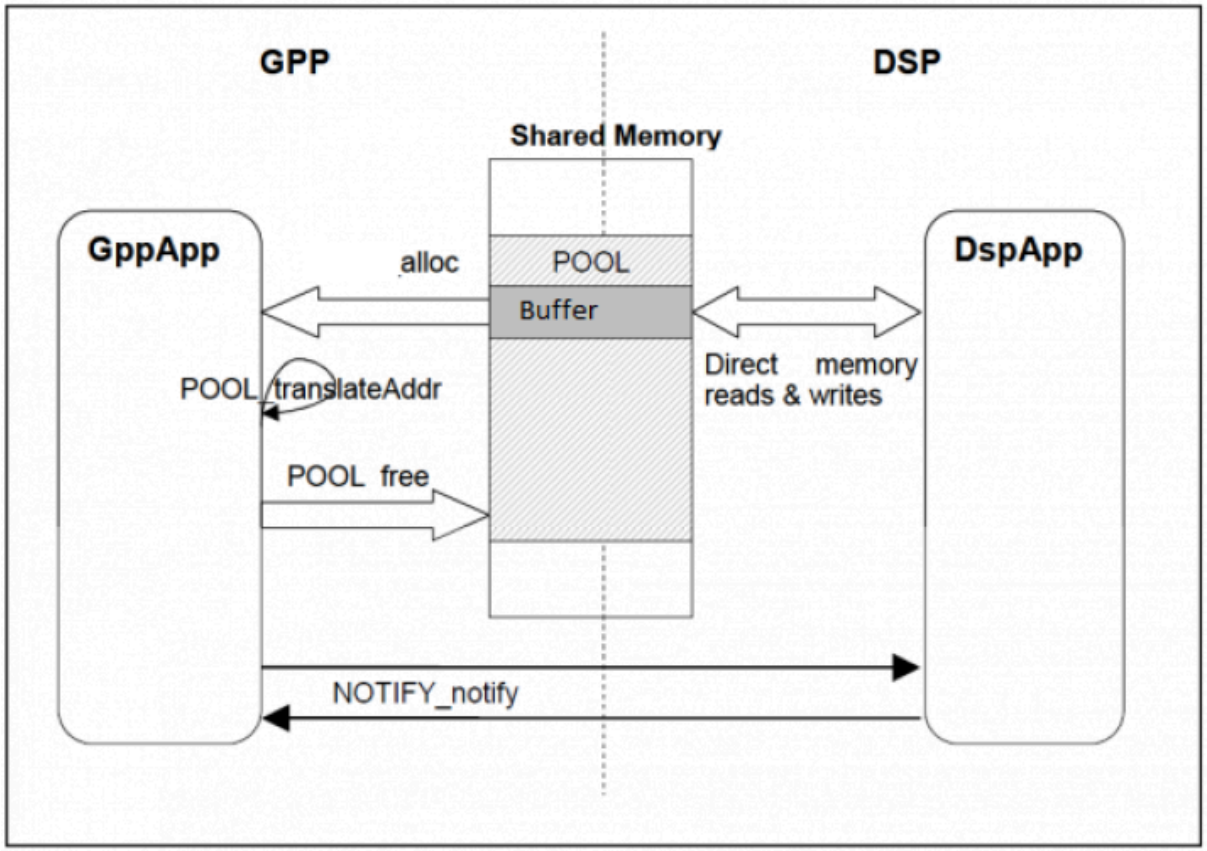


ADCS application
C code from Simulink

Yocto Project: Build System and Integration

Texas Instrument (DM3730 SoC)
ADCS OBC

Communication inside ADCS application



Optimized buffer size for EKF requirements

Integration Results

- EKF execution speedup of 5-10 times (based on baseline performance for FFT)
- ADCS software footprint size is less than 20 MB (regular footprint is 100-200 MB)
- Memory size for data exchange between ARM-DSP was reduced to 64MB (initially 128 MB)
- Fully automated operating system image generation with Yocto project + Jenkins

Conclusions

- “Y approach ” was introduced and implemented for DelFFi ADCS software architecture exploration
- Model profiling technique helped to identify and quantify computing demand for ADCS attitude estimation algorithm.
- Using a build system (Yocto project) and continuous integration tools improved software productivity problem

Further Work

- Continue the work in the porting of Simulink model to BeagleBoard XM board
- Compare ADCS performance with results from Flight Model OBC (Benchmark)
- Continue to investigate on code acceleration in space software applications with heterogeneous onboard processors

Thank You for your attention!!!

j.carvajalgodinez@tudelft.nl