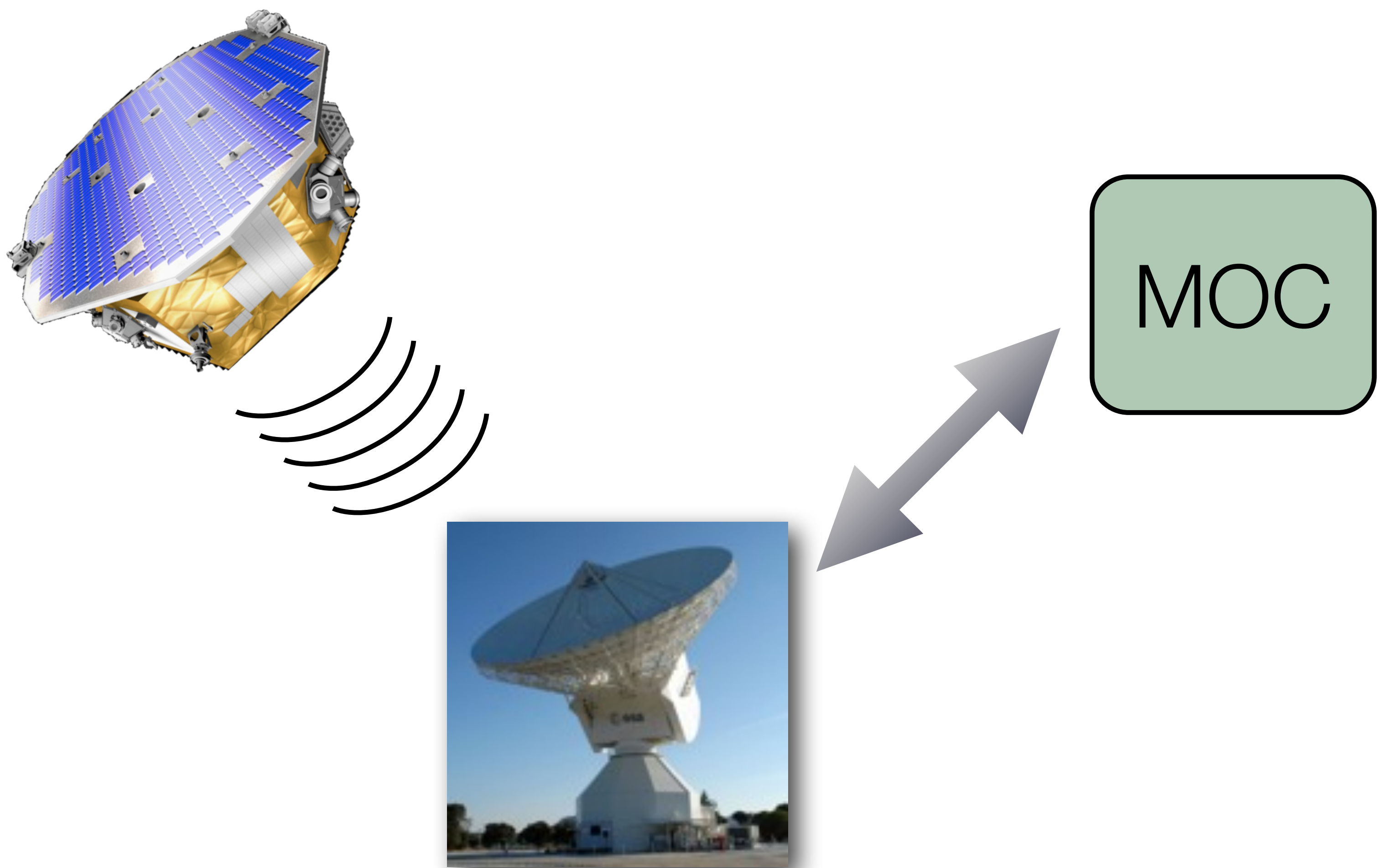
LPF Operations Heritage for LISA

M Hewitson for the LPF team
eLISA Consortium Meeting #1, APC, 22nd October 2012

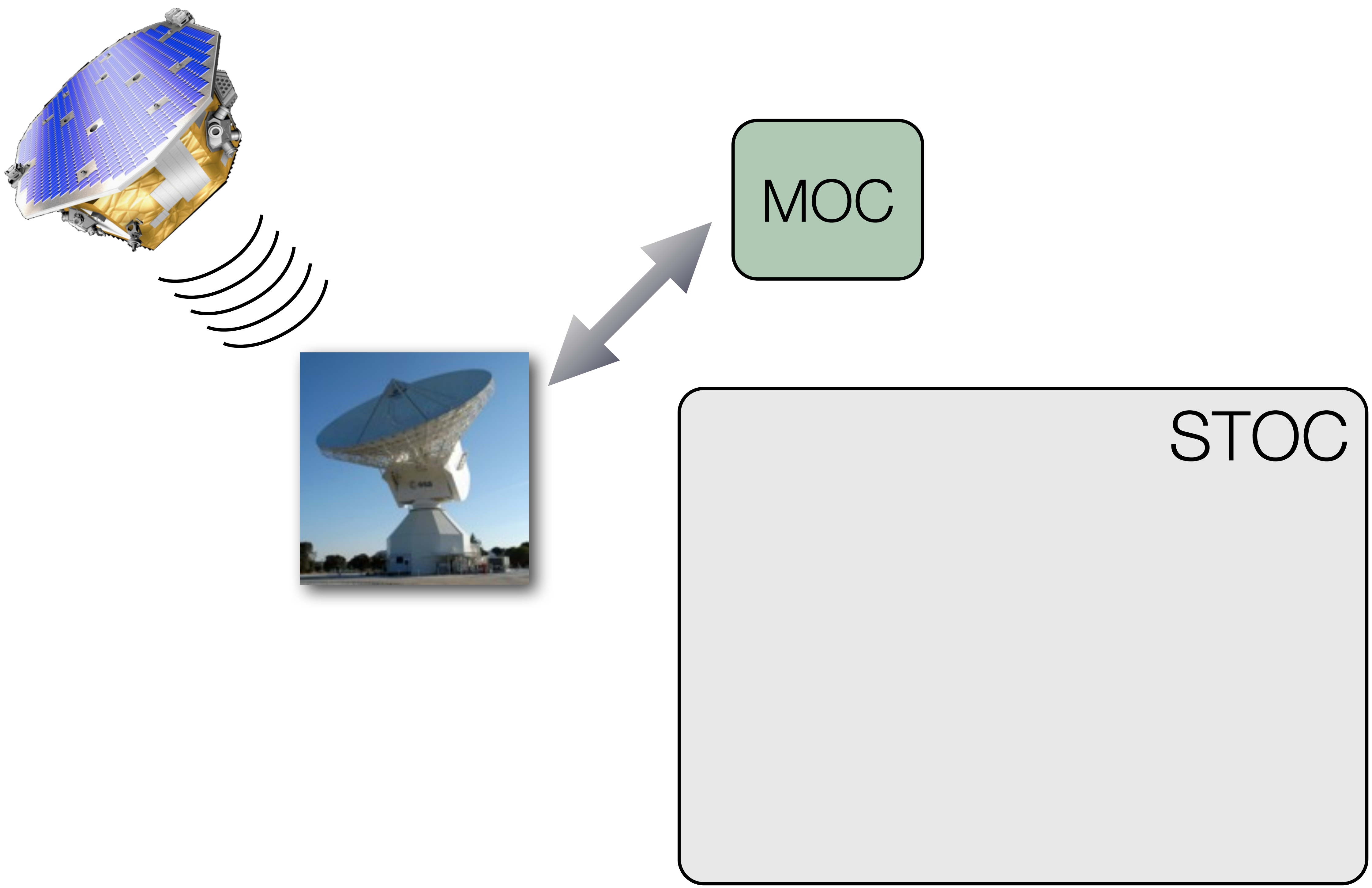
Contents

- Overview of LPF operations
- How this may fit to LISA
- Lessons (to be) learned

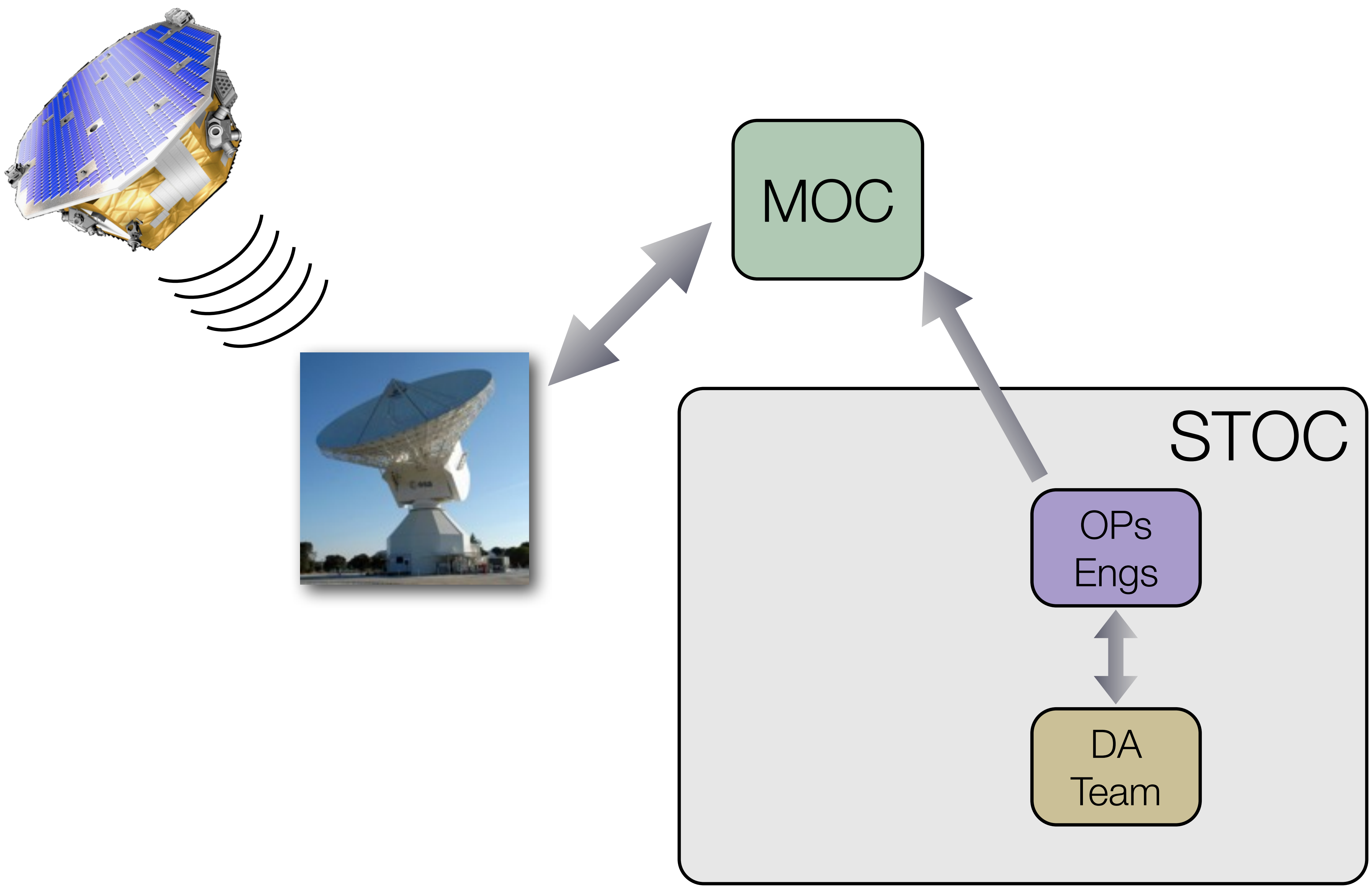
Structure of LPF operations (simplified)



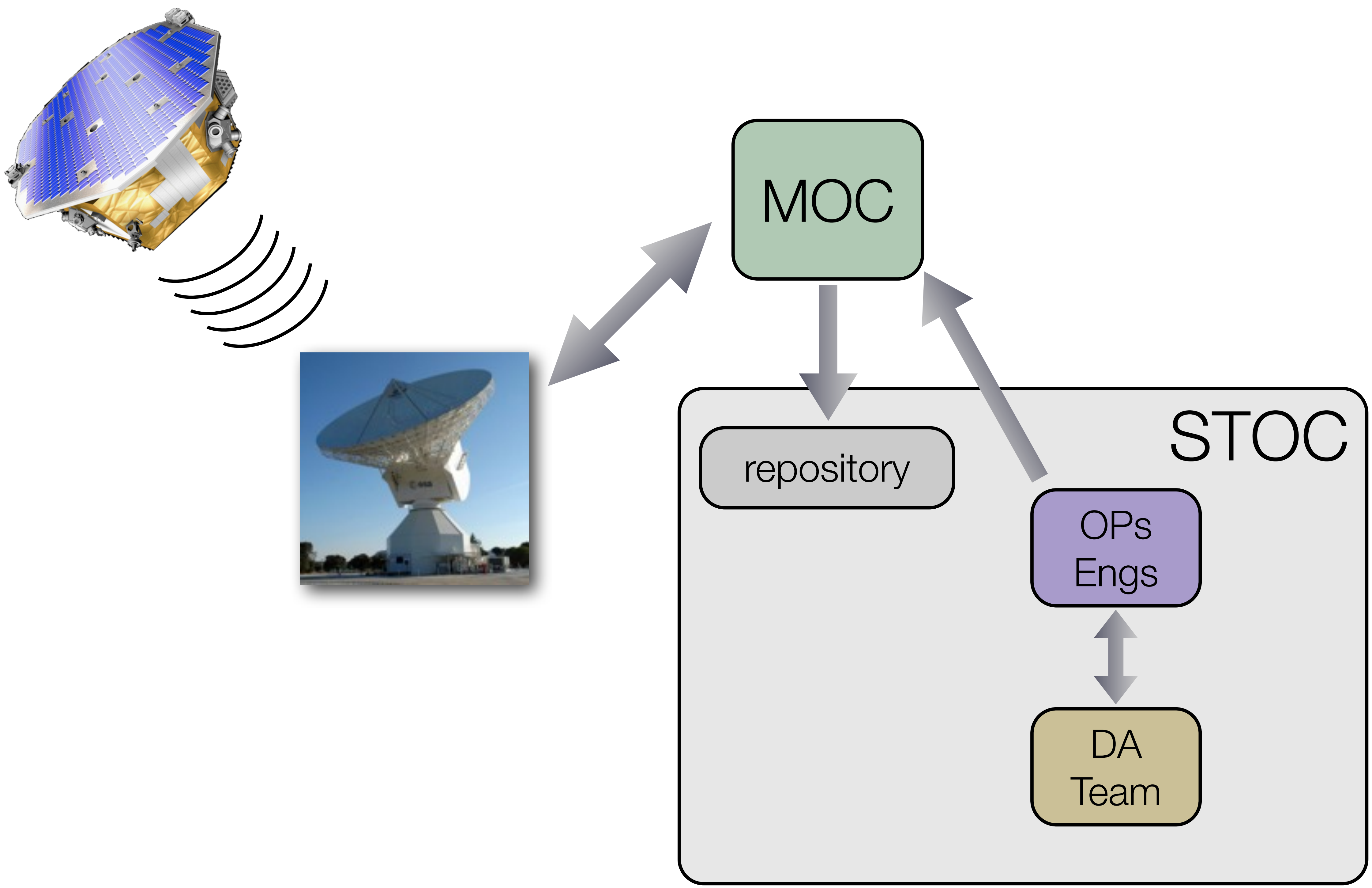
Structure of LPF operations (simplified)



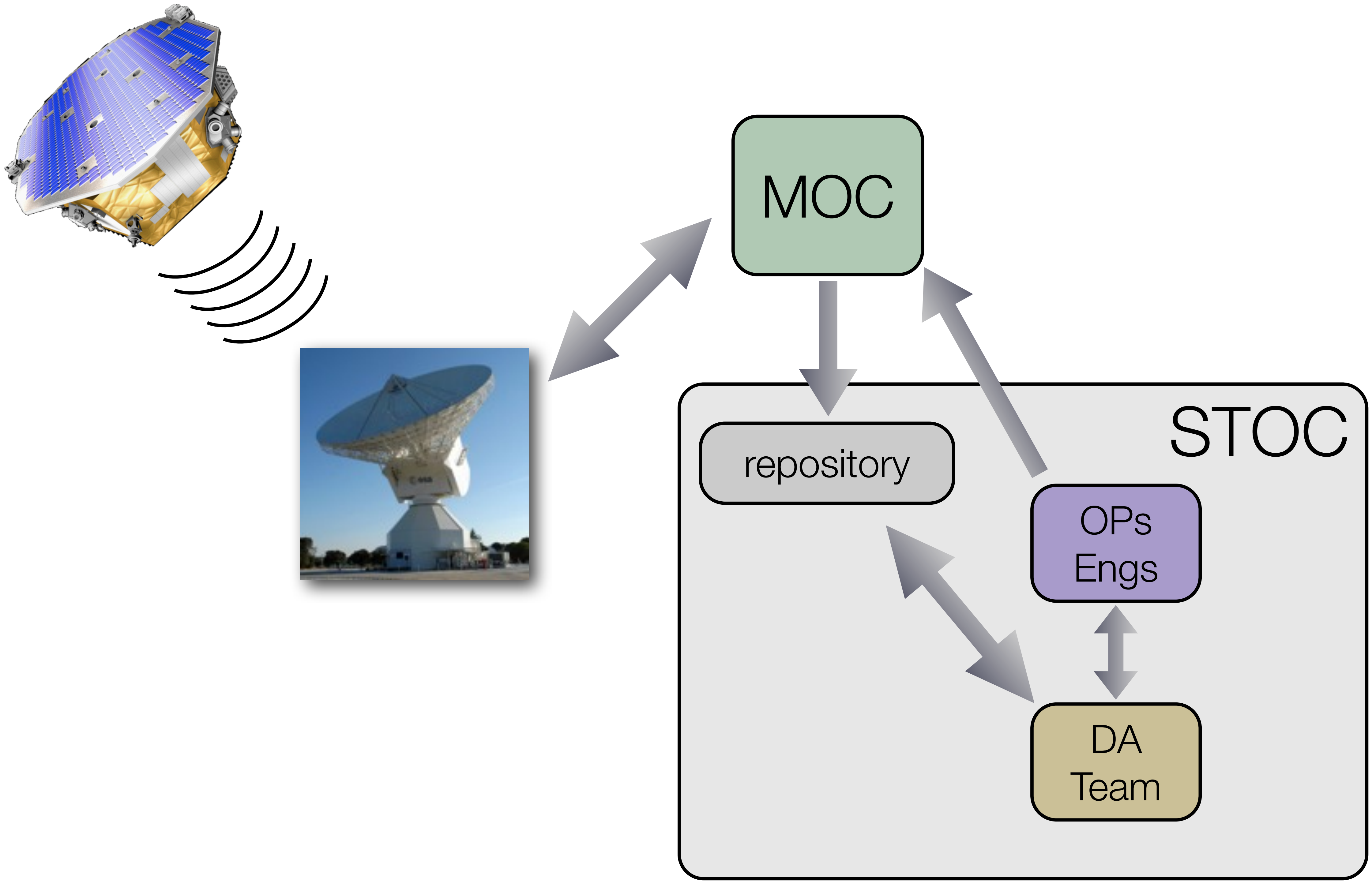
Structure of LPF operations (simplified)



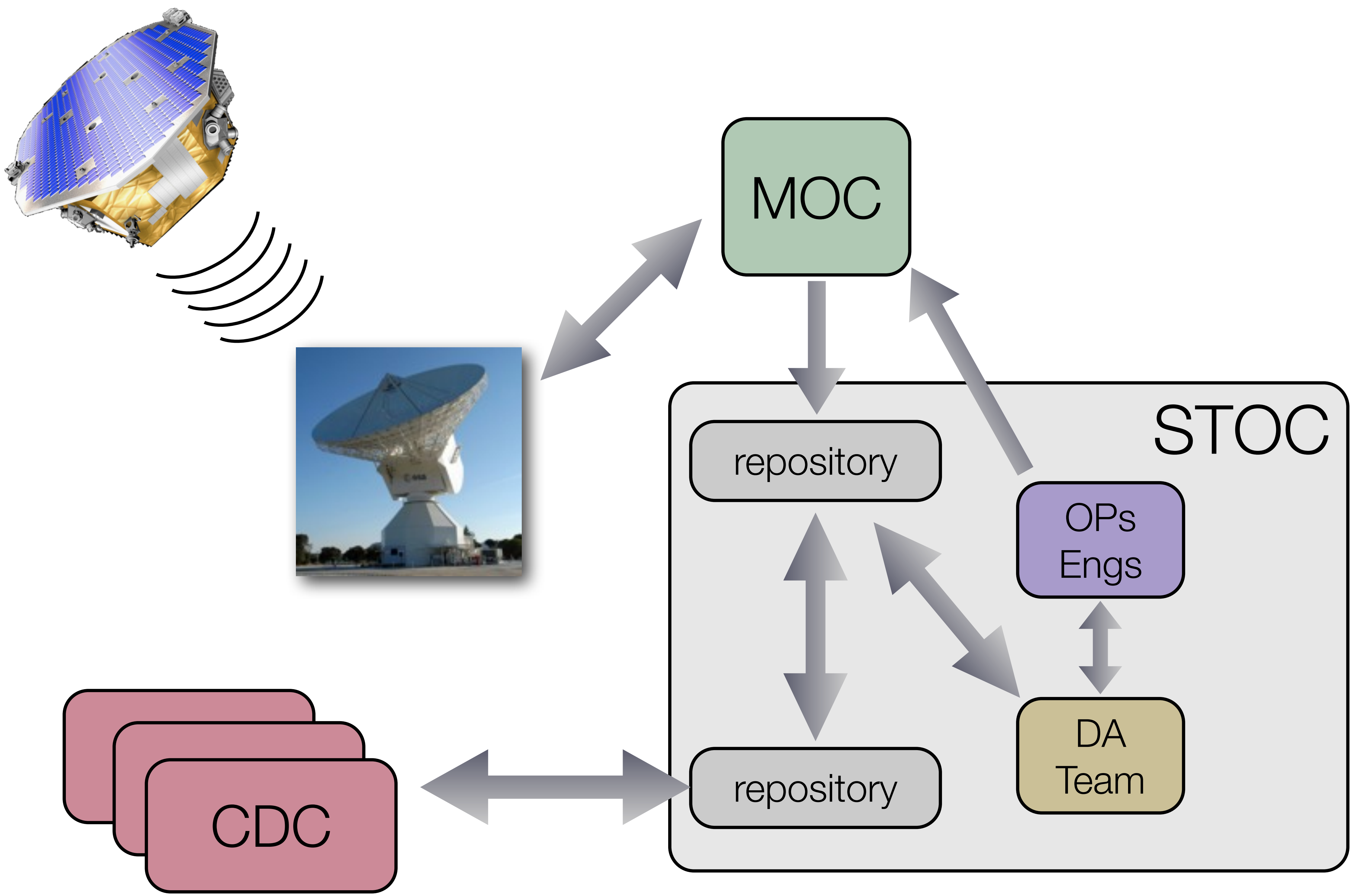
Structure of LPF operations (simplified)



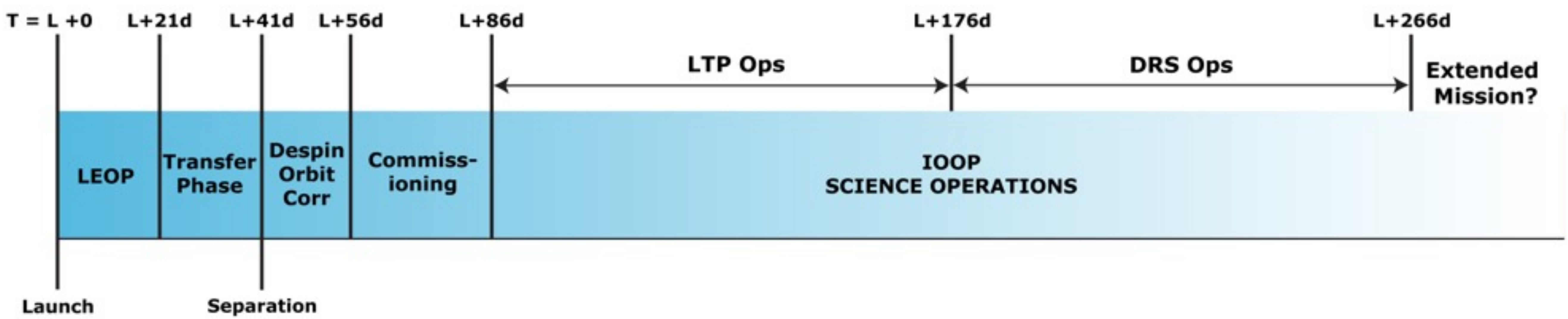
Structure of LPF operations (simplified)



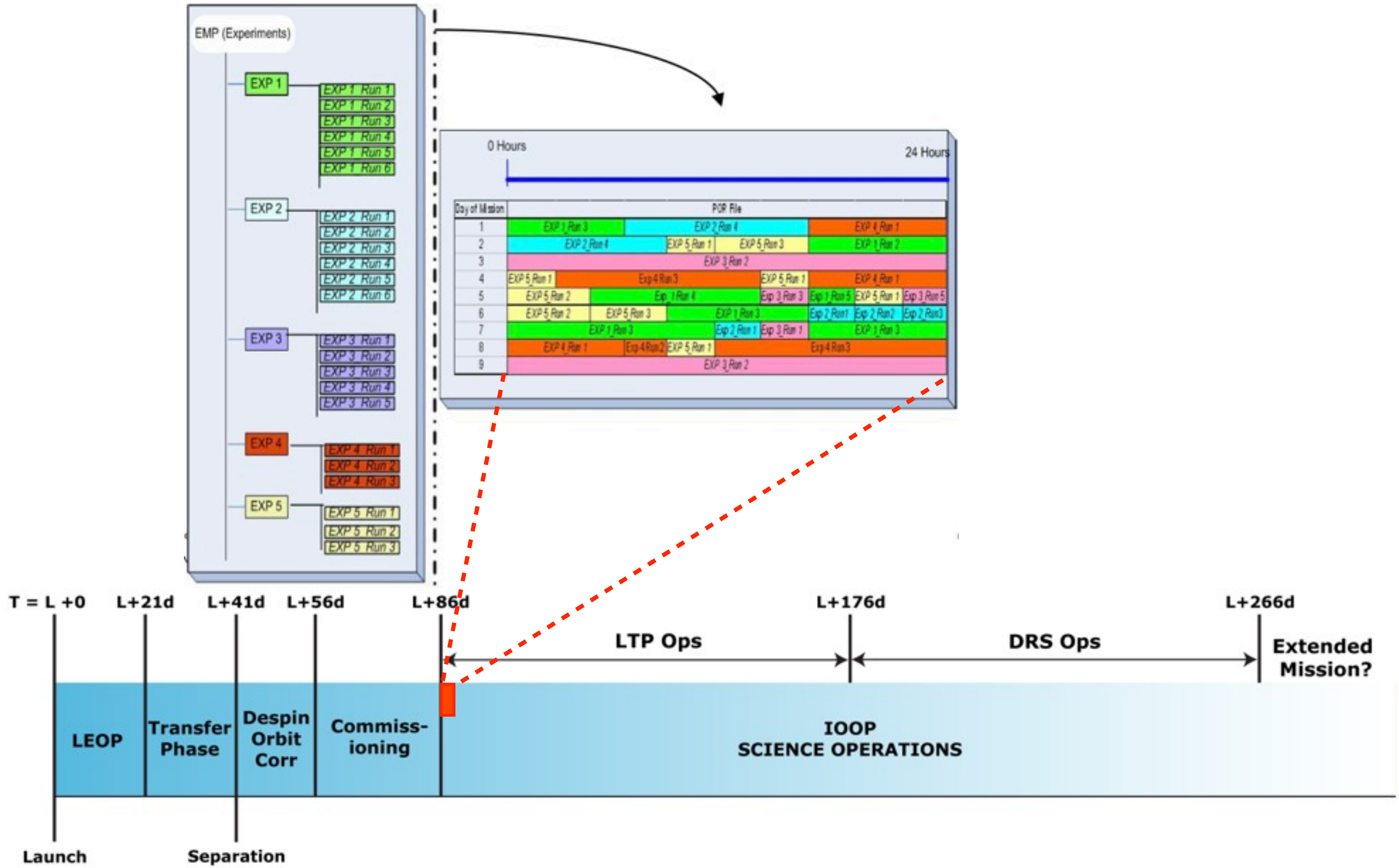
Structure of LPF operations (simplified)



Operations timing



Operations timing



Infrastructure

- Data handling
- Data analysis
- Documentation
- Simulators

Data Handling and Storage

Data Handling and Storage

- Data is received in packets from SC at MOC
- Converted to engineering values
  - simple calibrations applied
- Pushed into database

Data Handling and Storage

- Data is received in packets from SC at MOC
- Converted to engineering values
  - simple calibrations applied
- Pushed into database
- STOC pulls the data periodically
  - twice per day: high-priority data then consolidated

Data Handling and Storage

- Data is received in packets from SC at MOC
- Converted to engineering values
  - simple calibrations applied
- Pushed into database
- STOC pulls the data periodically
  - twice per day: high-priority data then consolidated
- Data is converted to LTPDA objects and pushed into LTPDA repository

Data Handling and Storage

- Data is received in packets from SC at MOC
- Converted to engineering values
  - simple calibrations applied
- Pushed into database
- STOC pulls the data periodically
  - twice per day: high-priority data then consolidated
- Data is converted to LTPDA objects and pushed into LTPDA repository
- From there data is pushed to external LTPDA repository where external scientists can access it

Data Handling and Storage

- Data is received in packets from SC at MOC
- Converted to engineering values
  - simple calibrations applied
- Pushed into database
- STOC pulls the data periodically
  - twice per day: high-priority data then consolidated
- Data is converted to LTPDA objects and pushed into LTPDA repository
- From there data is pushed to external LTPDA repository where external scientists can access it
- Data is also pushed to the ESA archive

Telemetry

- A huge number of ‘parameters’ are available on-board LPF
  - about 51,000 in all
  - typical science parameter set is about 100 parameters
- These all have names which need to be learned
- They all should have units and descriptions
  - part of the validation process to ensure this, though it’s typically the end user (DA team) who finds the problems
- We have a very limited band-width for downloading data
  - decide before-hand which sets of parameters to downlink

PCF_NAME	PCF_DESCR	PCF_DESCR2	PCF_PID	PCF_...	PCF_PTC	PCF_P...
ACT00001	CAC_CacsRateThreshCnm_0	Rate damping theshold before entry to CNM	31947	rs-1	5	2
ACT00002	CAC_CacsRateThreshCnm_1	Rate damping theshold before entry to CNM	31948	rs-1	5	2
ACT00003	CAC_CacsRateThreshCnm_2	Rate damping theshold before entry to CNM	31949	rs-1	5	2
ACT20001	CAC_CpsBFlg	Bitfield containing packed versions of CPS HK data	31955		3	14
ACT20002	CAC_CpsThrOntime_0	calculated on time for each thruster	31956	s	5	1
ACT20003	CAC_CpsThrOntime_1	calculated on time for each thruster	31957	s	5	1
ACT20004	CAC_CpsThrOntime_2	calculated on time for each thruster	31958	s	5	1
ACT20005	CAC_CpsThrOntime_3	calculated on time for each thruster	31959	s	5	1
ACT20006	CAC_CpsThrDelay_0	calculated delay time for each thruster	31960	s	5	1
ACT20007	CAC_CpsThrDelay_1	calculated delay time for each thruster	31961	s	5	1
ACT20008	CAC_CpsThrDelay_2	calculated delay time for each thruster	31962	s	5	1
ACT20009	CAC_CpsThrDelay_3	calculated delay time for each thruster	31963	s	5	1
ACT20010	CAC_CpsModRatio_0	current value of the modulation ratio	31964		5	1
ACT20011	CAC_CpsDeltaV_0	deltaV realised so far, in the commanded body frame, in CTCM	31968	ms	5	1
ACT20012	CAC_CpsDeltaV_1	deltaV realised so far, in the commanded body frame, in CTCM	31969	ms	5	1
ACT20013	CAC_CpsDeltaV_2	deltaV realised so far, in the commanded body frame, in CTCM	31970	ms	5	1
ACT20014	CAC_CpsAccumOntime_0	accumulated on time for each thruster	31971	s	5	1
ACT20015	CAC_CpsAccumOntime_1	accumulated on time for each thruster	31972	s	5	1
ACT20016	CAC_CpsAccumOntime_2	accumulated on time for each thruster	31973	s	5	1
ACT20017	CAC_CpsAccumOntime_3	accumulated on time for each thruster	31974	s	5	1
ACT20024	CAC_CpsThrAvailStatus	flag indicating availibility of the currently active CPS branch	31976		3	4
ACT20028	CAC_CpsMeAvailStatus	availability status of the main engine	31977		3	4
ACT20029	CAC_CpsMeOnOffStatus	on/off flag of the main engine - used to indicate a main engine fai...	31978		3	4
ACT20030	CAC_CpsBFlg_MeOnOffStBit	#ACT20029	0		2	1
ACT20031	CAC_CpsBFlg_MeAvailStBit	#ACT20028	0		2	1
ACT20032	CAC_CpsBFlg_ThrstAvlStBt	#ACT20024	0		2	1
ACT20055	CAC_CpsHkPadField1_0	Padding entry (not used)	31979		3	4
ACT20057	CAC_CpsModRatio_1	current value of the modulation ratio	31965		5	1
ACT20058	CAC_CpsModRatio_2	current value of the modulation ratio	31966		5	1
ACT20059	CAC_CpsModRatio_3	current value of the modulation ratio	31967		5	1
ACT20061	CAC_CpsCtotCount	Counter of CPS total on-time.	31975	s	5	1
ACT21001	CAC_CsamSubmode	the CSAM submode	32005		4	14
ACT21002	CAC_CsamSecondScanLaw	counter of times scan law has been entered	32007		3	4
ACT21003	CAC_CsamAuthoriseSu	flag to indicate if spin up has been authorised	32008		3	4
ACT21004	CAC_CsamAuthoriseCnm	flag to indicate if transition to CNM has been authorised	32009		3	4
ACT21005	CAC_CsamRdWait	time spent in CSAM rate damping submode	31983	s	5	1
ACT21006	CAC_CsamSlWait	time spent in CSAM scan law sunbode	31984	s	5	1
ACT21007	CAC_CsamEspWait	time spent in CSAM eclipse sun pointing sunbode	31985	s	5	1
ACT21008	CAC_CsamSuWait	time spent in CSAM spin up submode	31986	s	5	1
ACT21009	CAC_CsamRateRef_0	The reference rate	31987	rs-1	5	1
ACT21010	CAC_CsamRateRef_1	The reference rate	31988	rs-1	5	1
ACT21011	CAC_CsamRateRef_2	The reference rate	31989	rs-1	5	1
ACT21012	CAC_CsamRateErr_0	The rate error	31990	rs-1	5	1
ACT21013	CAC_CsamRateErr_1	The rate error	31991	rs-1	5	1
ACT21014	CAC_CsamRateErr_2	The rate error	31992	rs-1	5	1
ACT21015	CAC_CsamRateEst_0	The estimated rate	31993	rs-1	5	1
ACT21016	CAC_CsamRateEst_1	The estimated rate	31994	rs-1	5	1
ACT21017	CAC_CsamRateEst_2	The estimated rate	31995	rs-1	5	1
ACT21018	CAC_CsamSunAngErr_0	The sun angle error	31996	rad	5	1
ACT21019	CAC_CsamSunAngErr_1	The sun angle error	31997	rad	5	1
ACT21020	CAC_CsamSunAngEst_0	The estimated sun sngles	31998	rad	5	1
ACT21021	CAC_CsamSunAngEst_1	The estimated sun sngles	31999	rad	5	1
ACT21022	CAC_CsamSunAngRef_0	The reference sun angles	32000	rad	5	1
ACT21023	CAC_CsamSunAngRef_1	The reference sun angles	32001	rad	5	1
ACT21024	CAC_CsamBodyTorque_0	The calculated body torque	32002	Nm	5	1
ACT21025	CAC_CsamBodyTorque_1	The calculated body torque	32003	Nm	5	1
ACT21026	CAC_CsamBodyTorque_2	The calculated body torque	32004	Nm	5	1
ACT21027	CAC_CsamBFlg	Bitfield containing packed versions of CSAM HK data	32006		3	12
ACT21031	CAC_CsamCruiseFlag	flag to indicate if CSAM is currently in cruise phase	32010		3	4
ACT21032	CAC_CsamHkPadField1_0	Padding entry (not used)	32011		3	4
ACT21033	CAC_CsamHkPadField1_1	Padding entry (not used)	32012		3	4
ACT21034	CAC_CsamBFlg_SubmodeBit	#ACT21001	0		2	3
ACT21035	CAC_CsamBFlg_AuthSuBit	#ACT21003	0		2	1
ACT21036	CAC_CsamBFlg_AuthCnmBit	#ACT21004	0		2	1
ACT21037	CAC_CsamBFlg_2ndScnLwBit	#ACT21002	0		3	4
ACT22001	CAC_CnmBodyTorque_0	The calculated body torque	32016	Nm	5	1
ACT22002	CAC_CnmBodyTorque_1	The calculated body torque	32017	Nm	5	1
ACT22003	CAC_CnmBodyTorque_2	The calculated body torque	32018	Nm	5	1
ACT22004	CAC_CnmCruiseFlag	flag to indicate if CNM is currently in cruise phase	32019		3	4

Data Repository

- The LPF DA repository is a database backend (MySQL) with a web-interface and a MATLAB client interface
- Parameters are stored in discrete chunks of time
- We will generate many objects
  - raw data and analysis products
- Good meta-data for searching is essential


```
>> a = ao(plist('hostname', '130.7
M:                constructing fr
M:        running ao/ao
M:    constructing from plist
M:      constructing from repository 130.75.117.67
M:    connection id=1 closed
M:    use cached credentials
M:    connection to mysql://130.75.117.67/ltpda_test username=hewitson
M:    cache password
M:    add connection to pool
M:  running ltpda_uo/retrieve
M:    binary retrieve
M:    retrieving objects 604809
M:    updated transactions table
M:        running ao/display
----------- ao 01: My test: objmeta -----------

       name: My test: objmeta
       data: (1,-0.401021818931651) (2,-0.950917208714911) (3,0.158083762190679)
(4,0.838654812586061) (5,-0.0682957451436672) ...
```

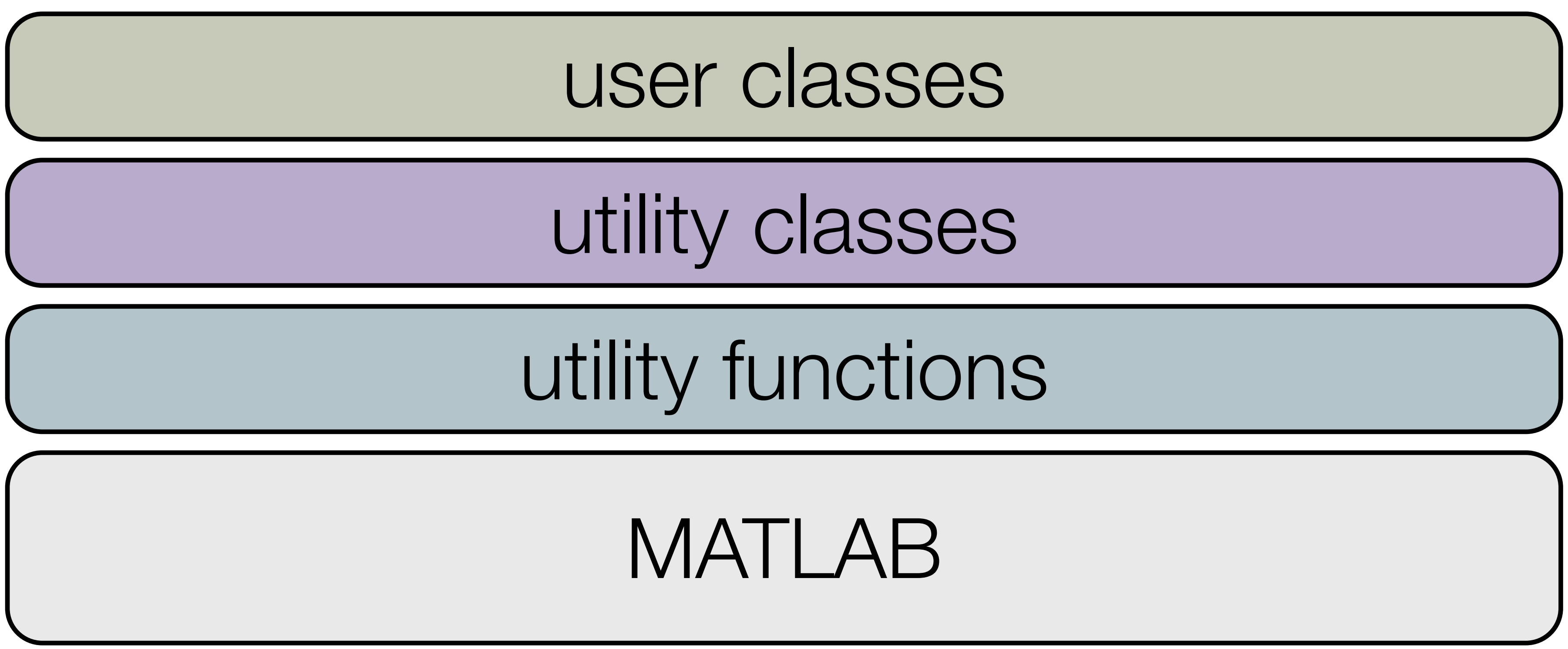
Data Analysis Tools

- toolbox algorithms
- collection of pipelines
- interface with data repository
- interface with mission database

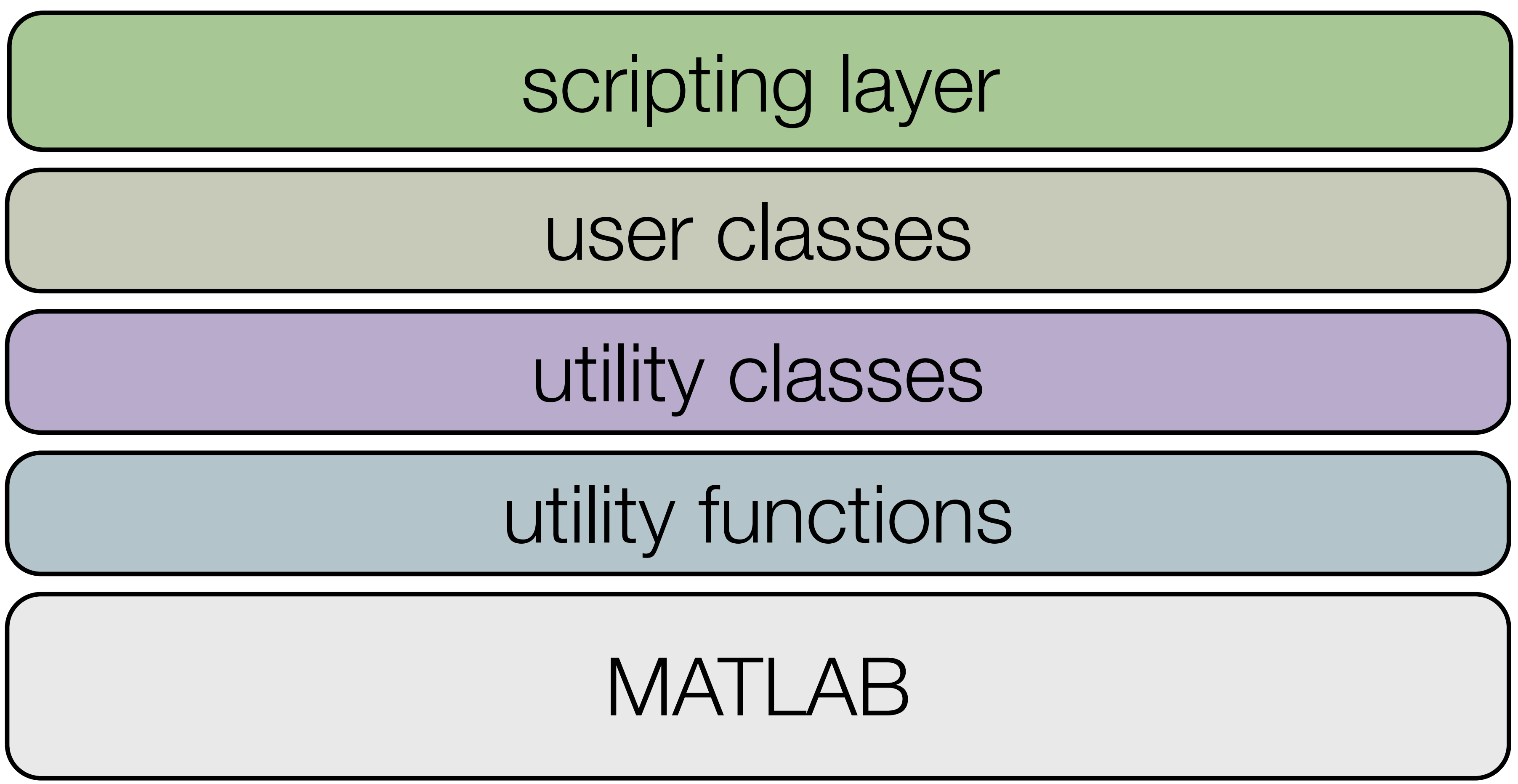
LTPDA Toolbox

MATLAB

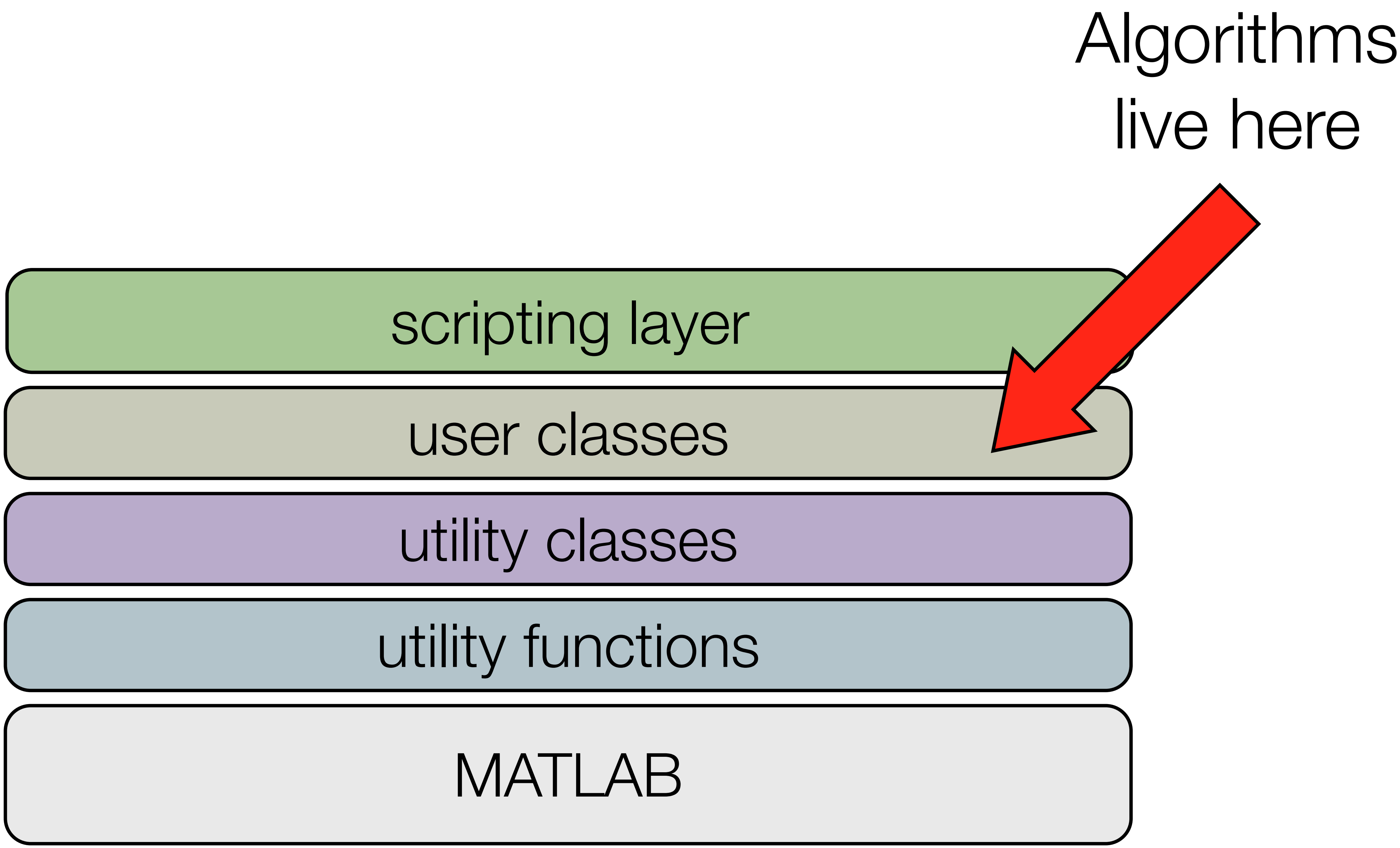
LTPDA Toolbox



LTPDA Toolbox

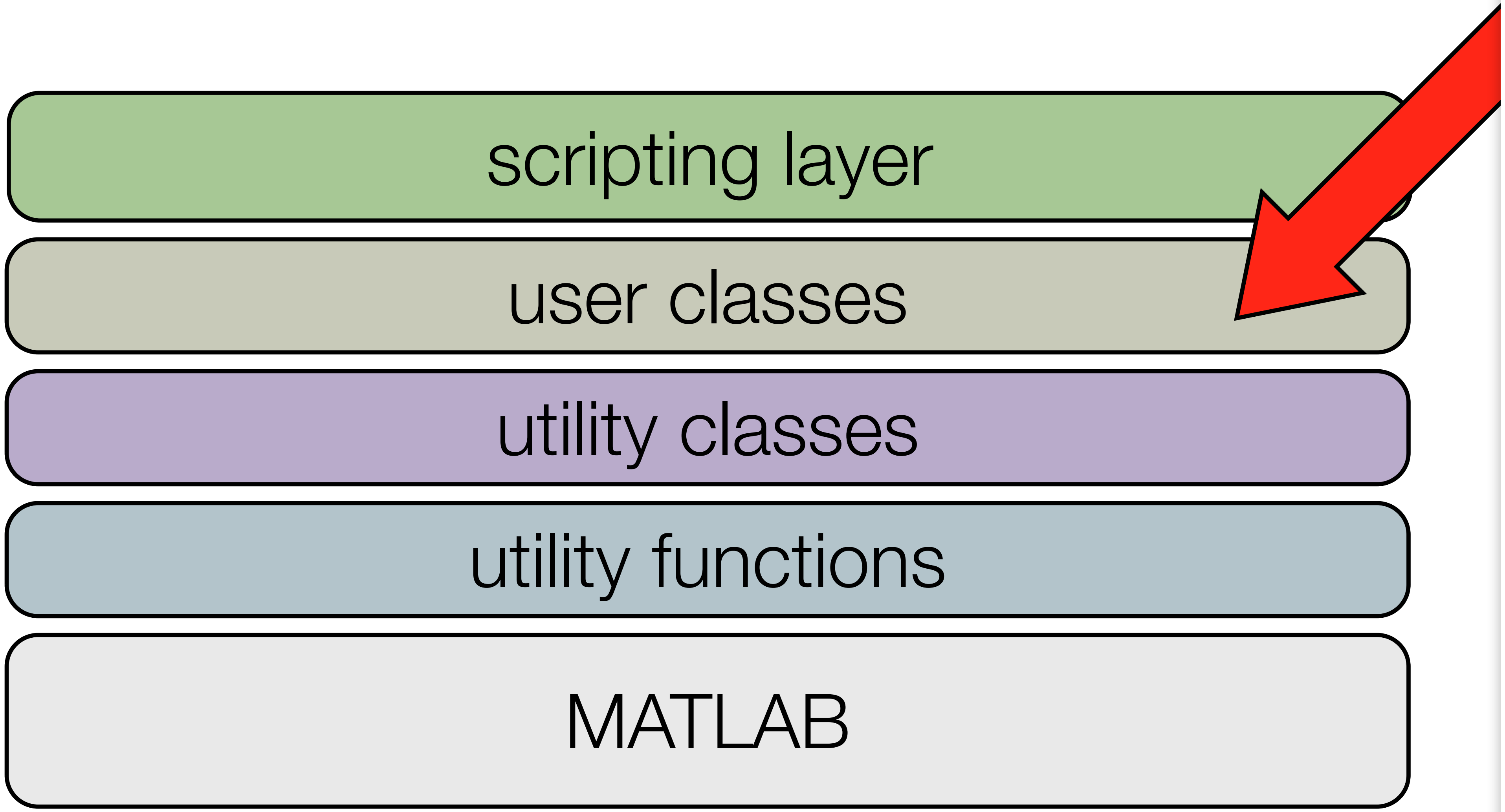


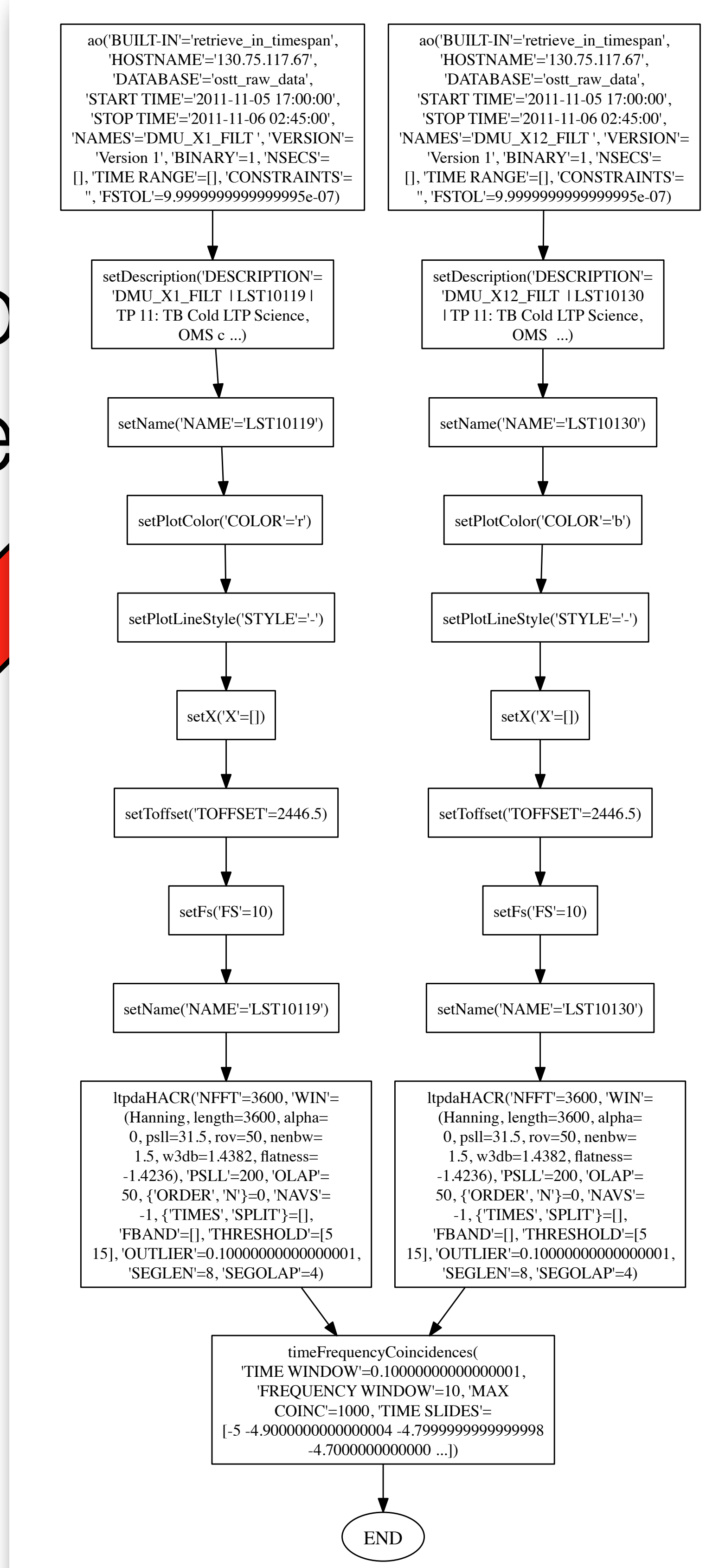
LTPDA Toolbox



LTPDA Toolbox

Algo
live



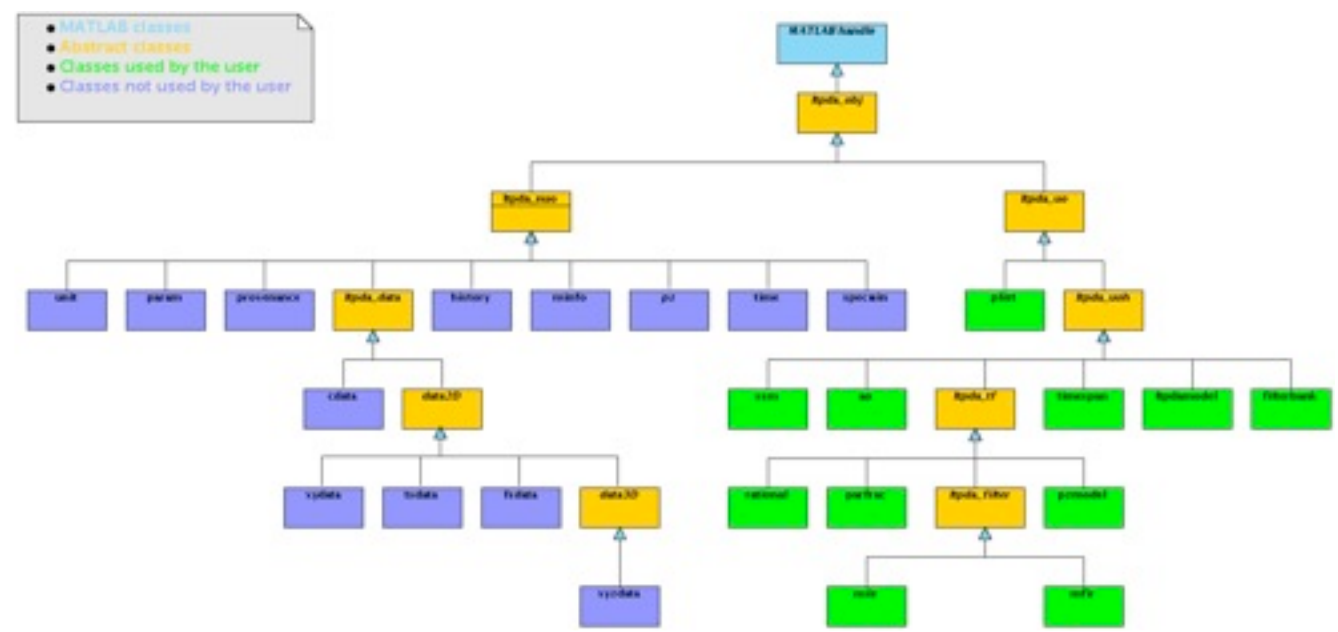


Tuesday, October 23, 12

Algorithms

- We have many classes of objects
- Each class of object has algorithms

Contents	fs	psdconf
abs	gapfilling	pwelch
acos	gapfillingoptim	rdivide
addlistener	ge	real
angle	get	rebuild
ao	getBuiltInModels	report
ao2m	getdof	resample
asin	gt	rms
atan	heterodyne	sDomainFit
atan2	hist	save
attachm	ifft	scale
attachmdl	imag	search
bsubmit	index	select
cat	interp	setDescription
char	interpmissing	setDx
cohere	inv	setDy
complex	iplot	setFs
compute	iplotyy	setName
confint	isprop	setPlotinfo
conj	isvalid	setProperties
consolidate	join	setT0
conv	lcohere	setX
convert	lcpsd	setXY
copy	le	setXunits
corr	len	setY
cos	lincom	setYunits
cov	linedetect	setZ
cpsd	lisovfit	sign
created	ln	simplifyYunits
creator	loadobj	sin
ctranspose	log	smoother



Spectral Estimates
PSD, Coherence, CPSD, TF

Noise subtraction
Time-series pre-processing
Digital filtering

Parameter Estimates
Signal extraction
model fitting

Noise budget
Transfer functions
noise sources
projection

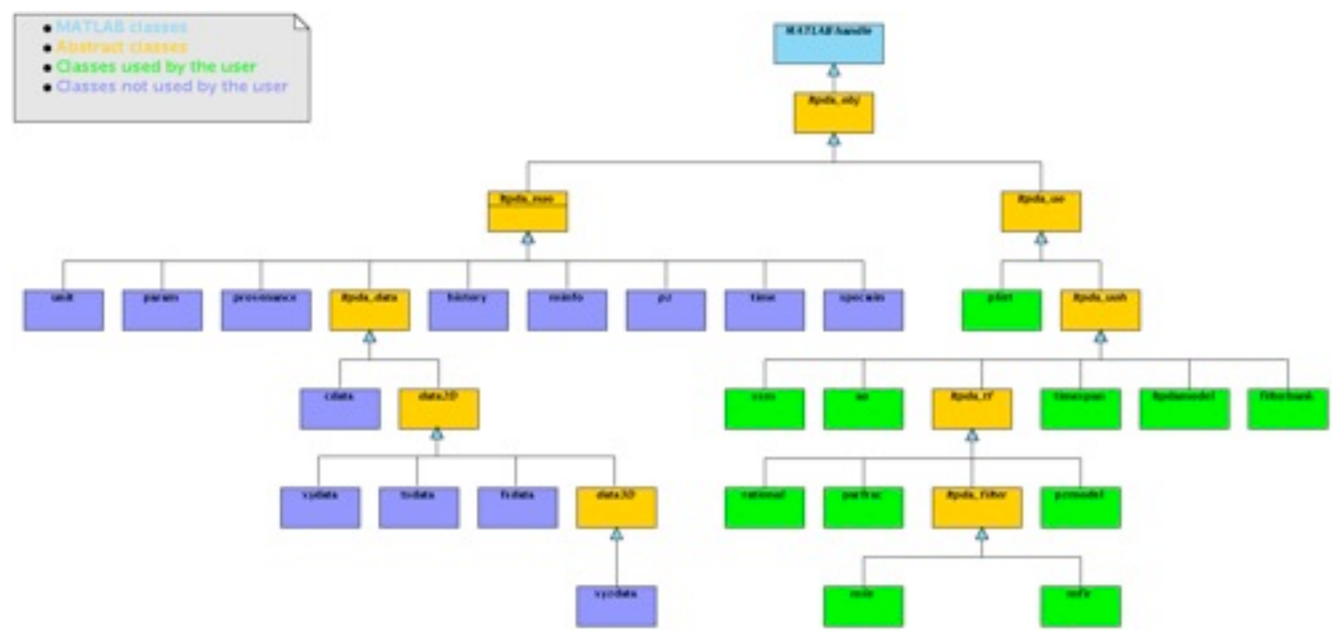
Calibrations
Conversion to $\ddot{x}$
Extract physical quantities

Simulation
Template fitting
Dynamical responses
Analytical models

Contents
abs
acos
addlistener
angle
ao
ao2m
asin
atan
atan2
attachm
attachmdl
bsubmit
cat
char
cohere
complex
compute
confint
conj
consolidate
conv
convert
copy
corr
cos
cov
cpsd
created
creator
ctranspose
curvefit
curvefit2
delay
delete
demux
det
detrend
dft
diag
diff
display
dopplercorr
downsample
dropduplicates
dsmean
dx
dy
fs
gapfilling
gapfillingoptim
ge
get
getBuiltInModels
getdof
gt
heterodyne
hist
ifft
imag
index
interp
interpmissing
inv
iplot
iplotyy
isprop
isvalid
join
lcohere
lcpsd
le
len
lincom
linedetect
lisovfit
ln
loadobj
log
log10
lpsd
lscov
lt
ltfe
max
md5
mdc1_cont2act_utn
mdc1_ifo2acc_fd
mdc1_ifo2acc_fd_utn
mdc1_ifo2acc_inloop
mdc1_ifo2cont_utn
mdc1_ifo2control
mdc1_x2acc
mean
median
metropolis
psdconf
pwelch
rdivide
real
rebuild
report
resample
rms
sDomainFit
save
scale
search
select
setDescription
setDx
setDy
setFs
setName
setPlotinfo
setProperties
setT0
setX
setXY
setXunits
setY
setYunits
setZ
sign
simplifyYunits
sin
smoother
sort
spectrogram
spikecleaning
split
sqrt
std
straightLineFit
string
submit
sum
sumjoin
svd
t0
table
tan
tfe
timedomainfit

...ses of objects

...t has algorithms



Spectral Estimates
PSD, Coherence, CPSD, TF

Noise subtraction
Time-series pre-processing
Digital filtering

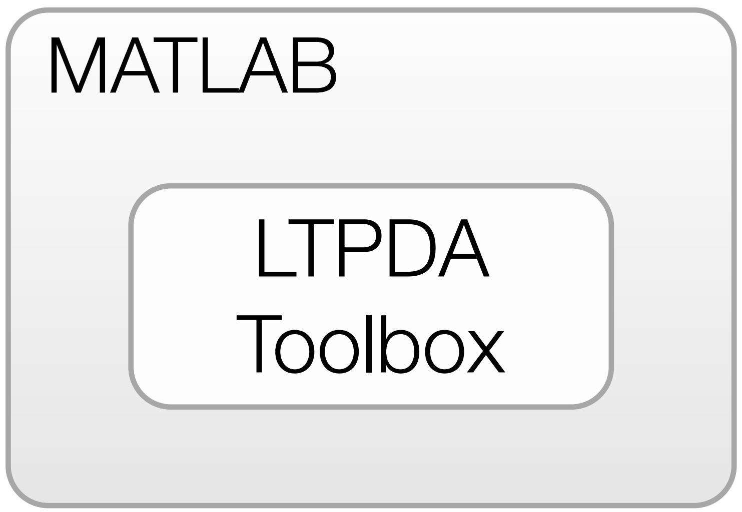
Parameter Estimates
Signal extraction
model fitting

Noise budget
Transfer functions
noise sources
projection

Calibrations
Conversion to $\ddot{x}$
Extract physical quantities

Simulation
Template fitting
Dynamical responses
Analytical models

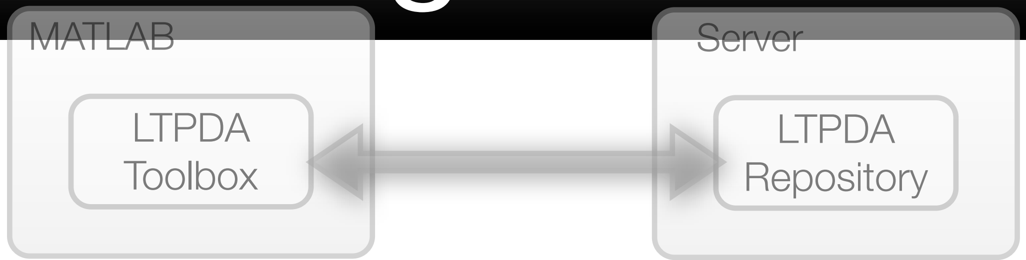
Getting Data



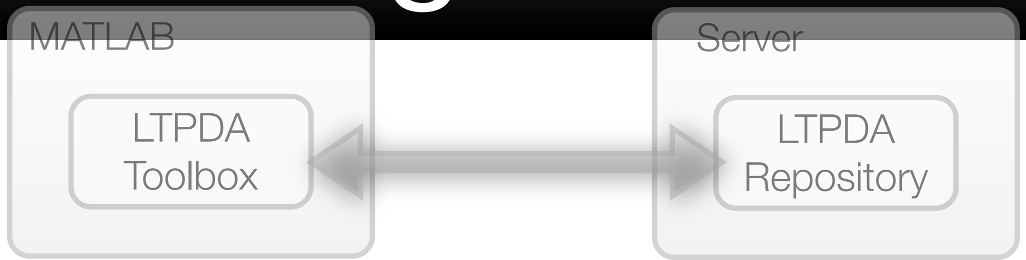
Getting Data

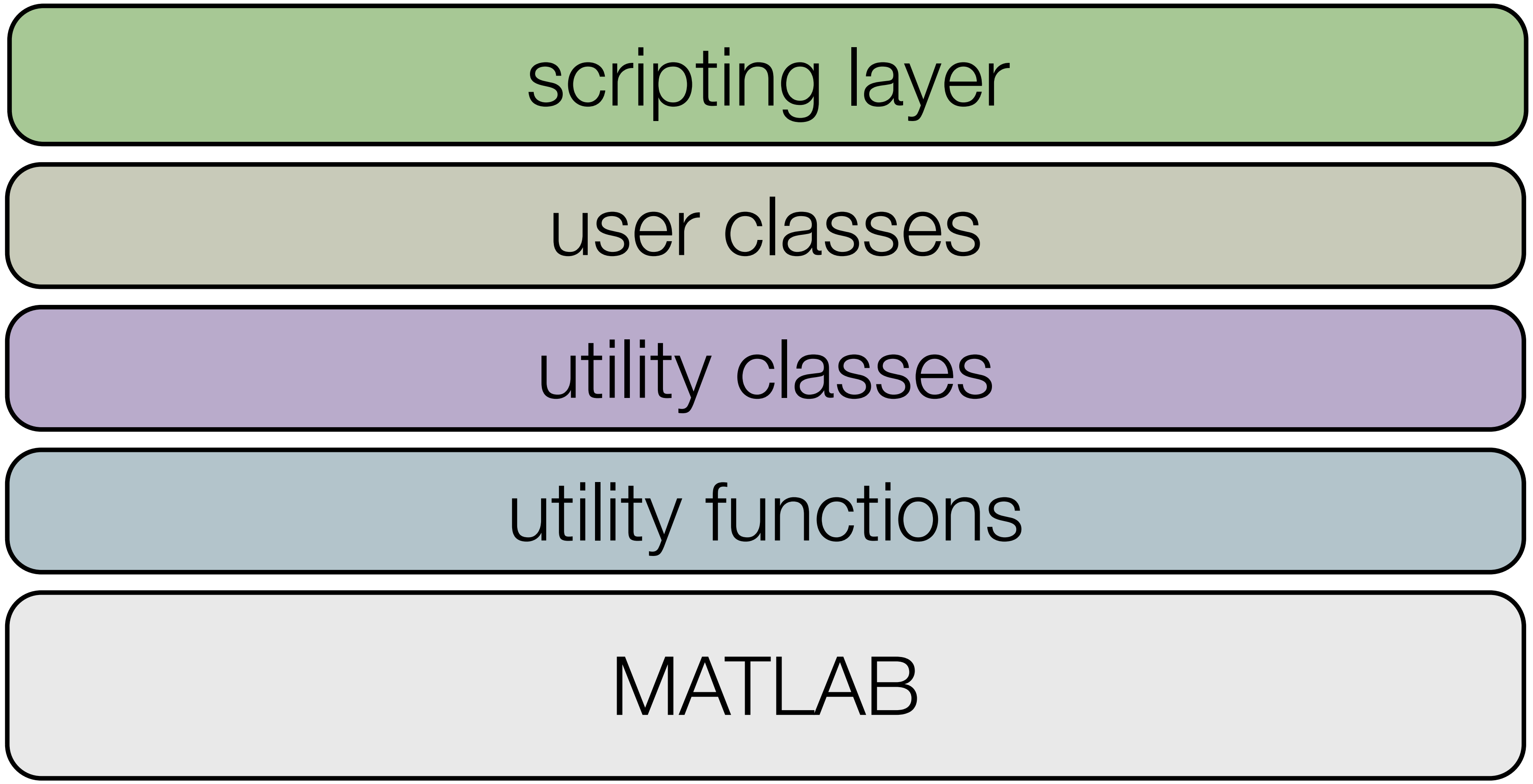
Getting Data

Getting Data

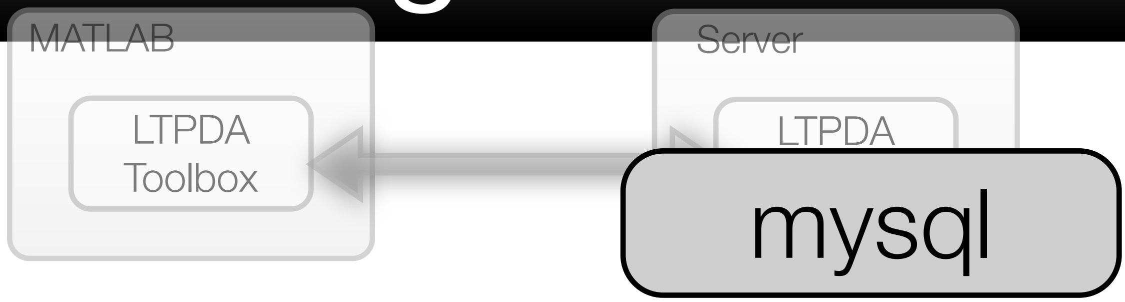


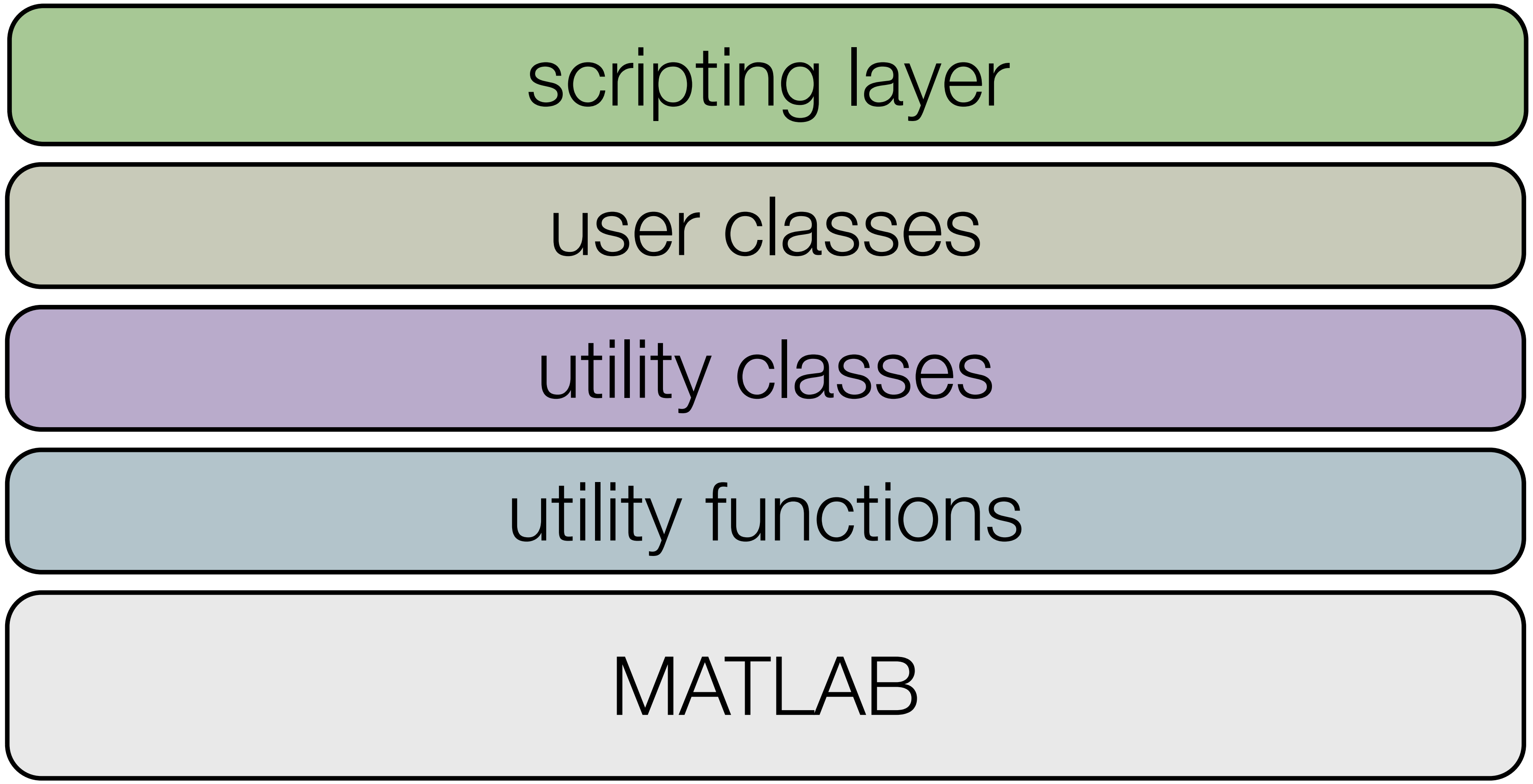
Getting Data



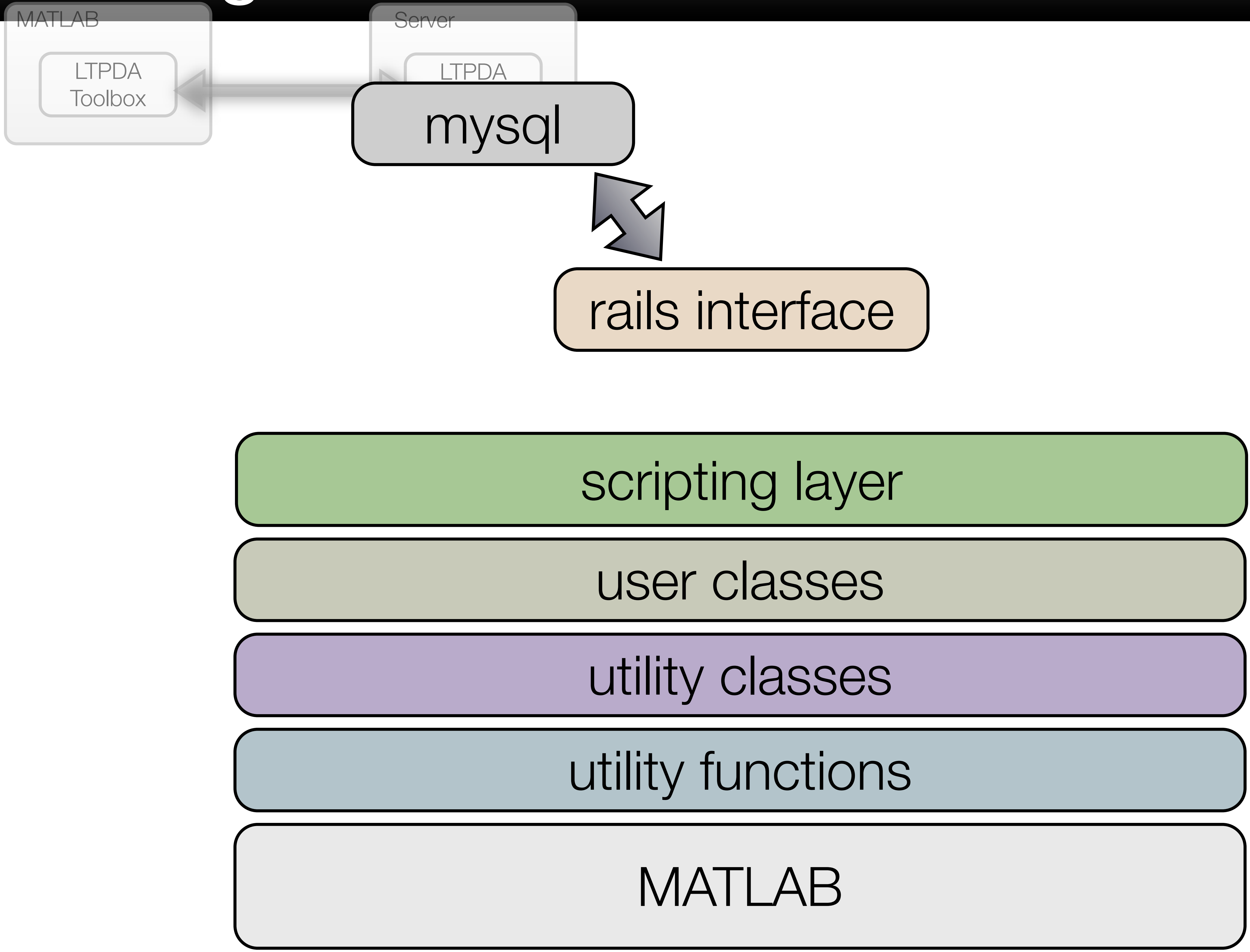


Getting Data

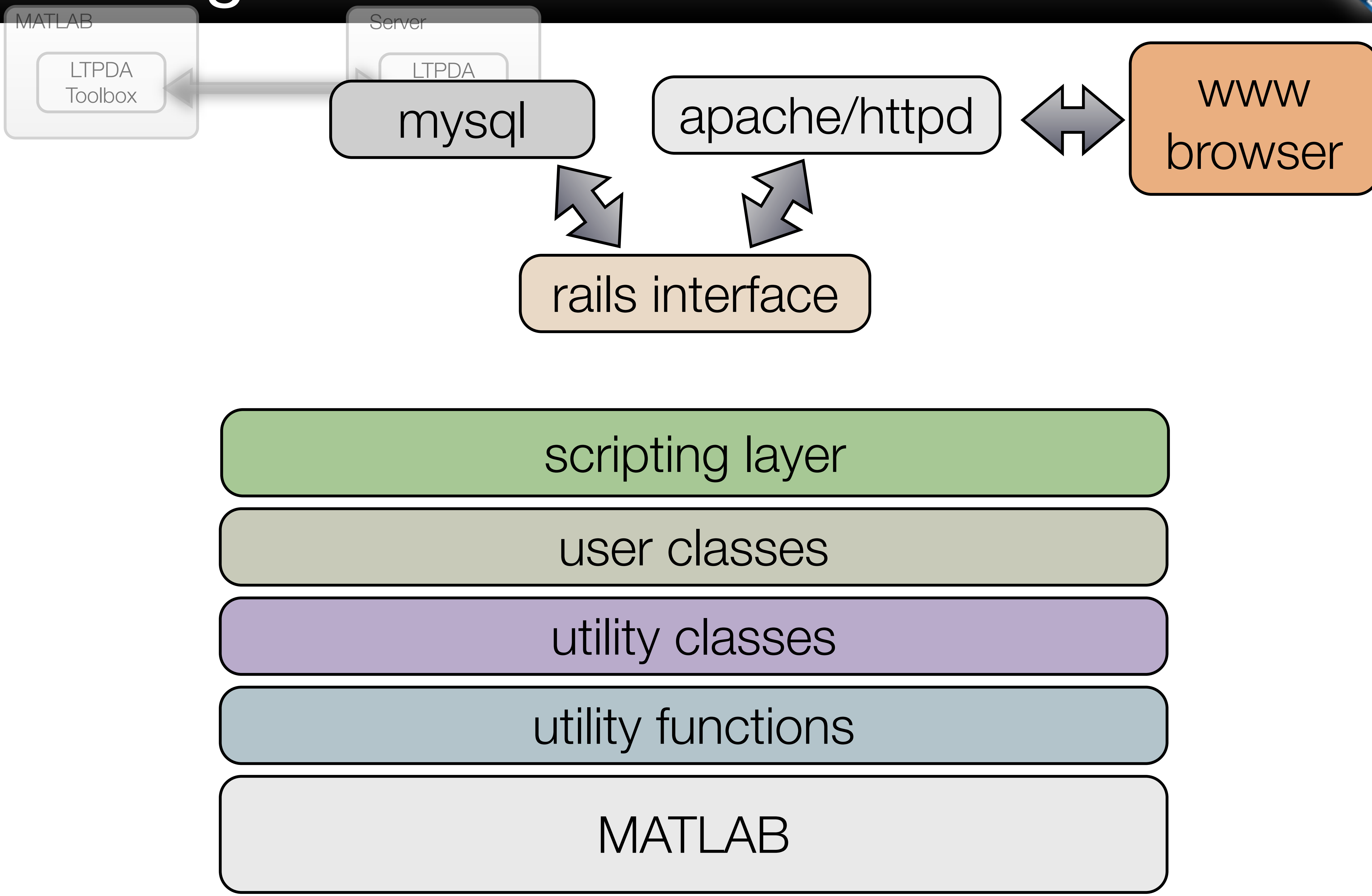




Getting Data

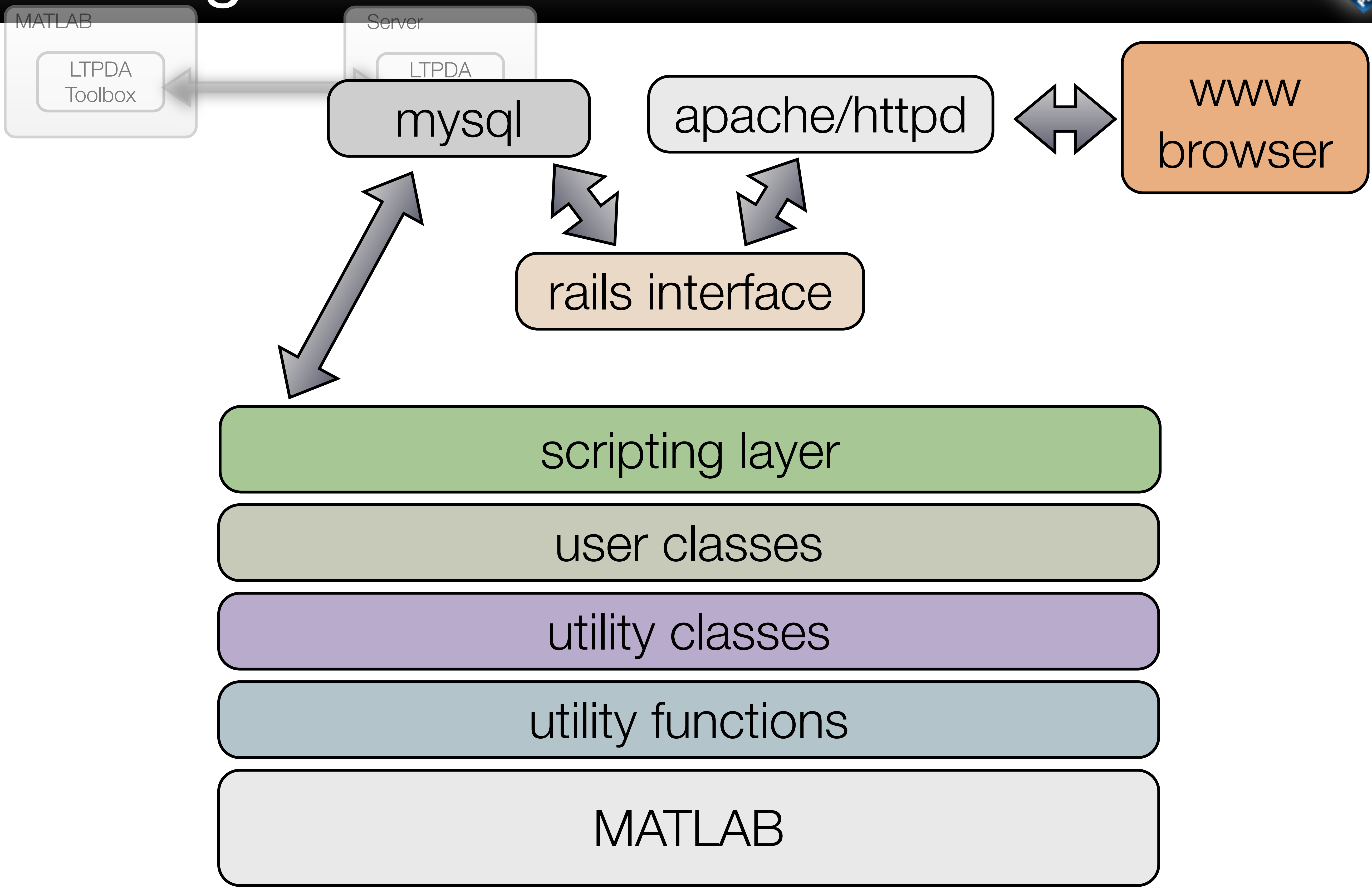


Getting Data



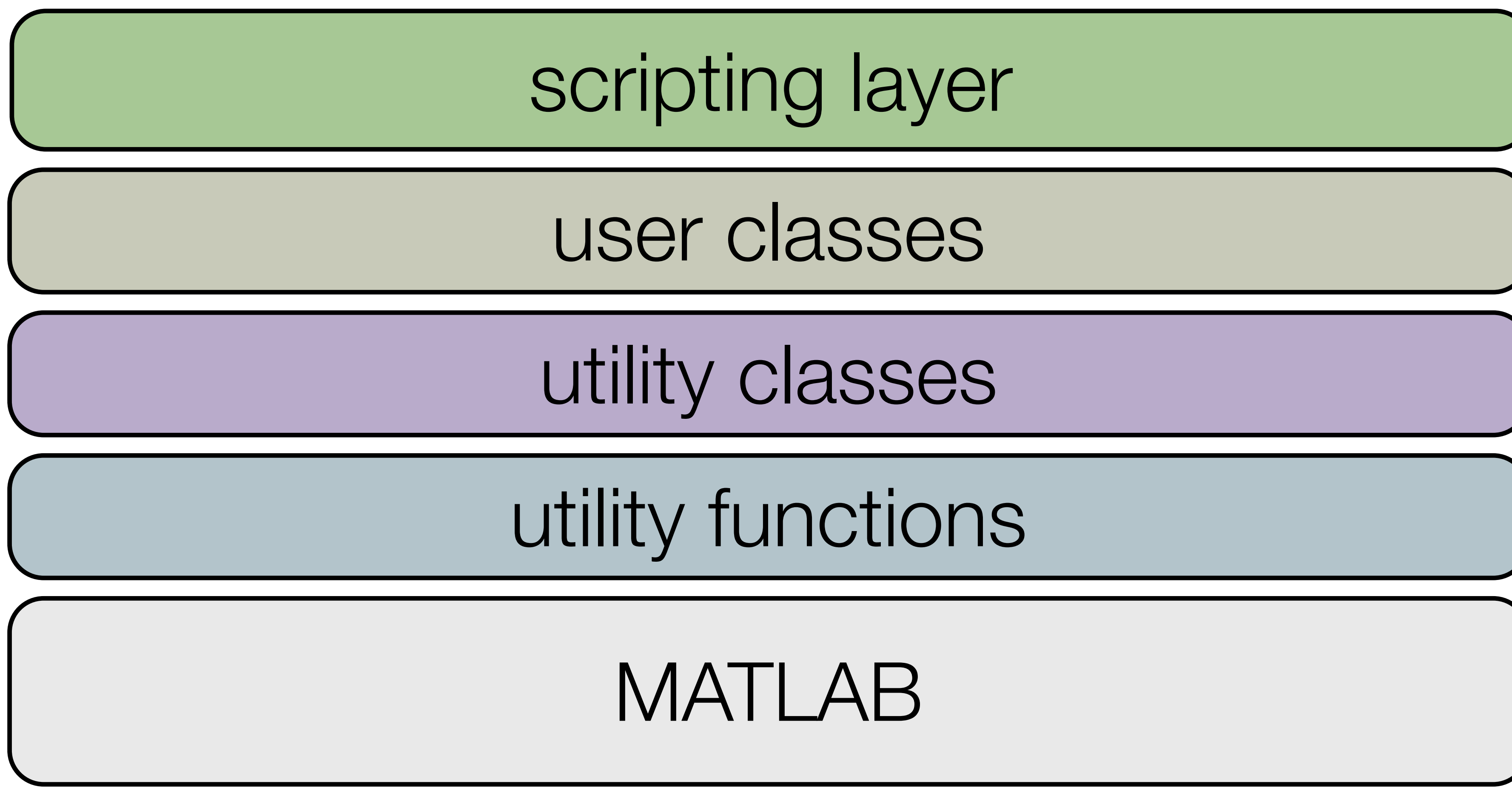
Tuesday, October 23, 12

Getting Data



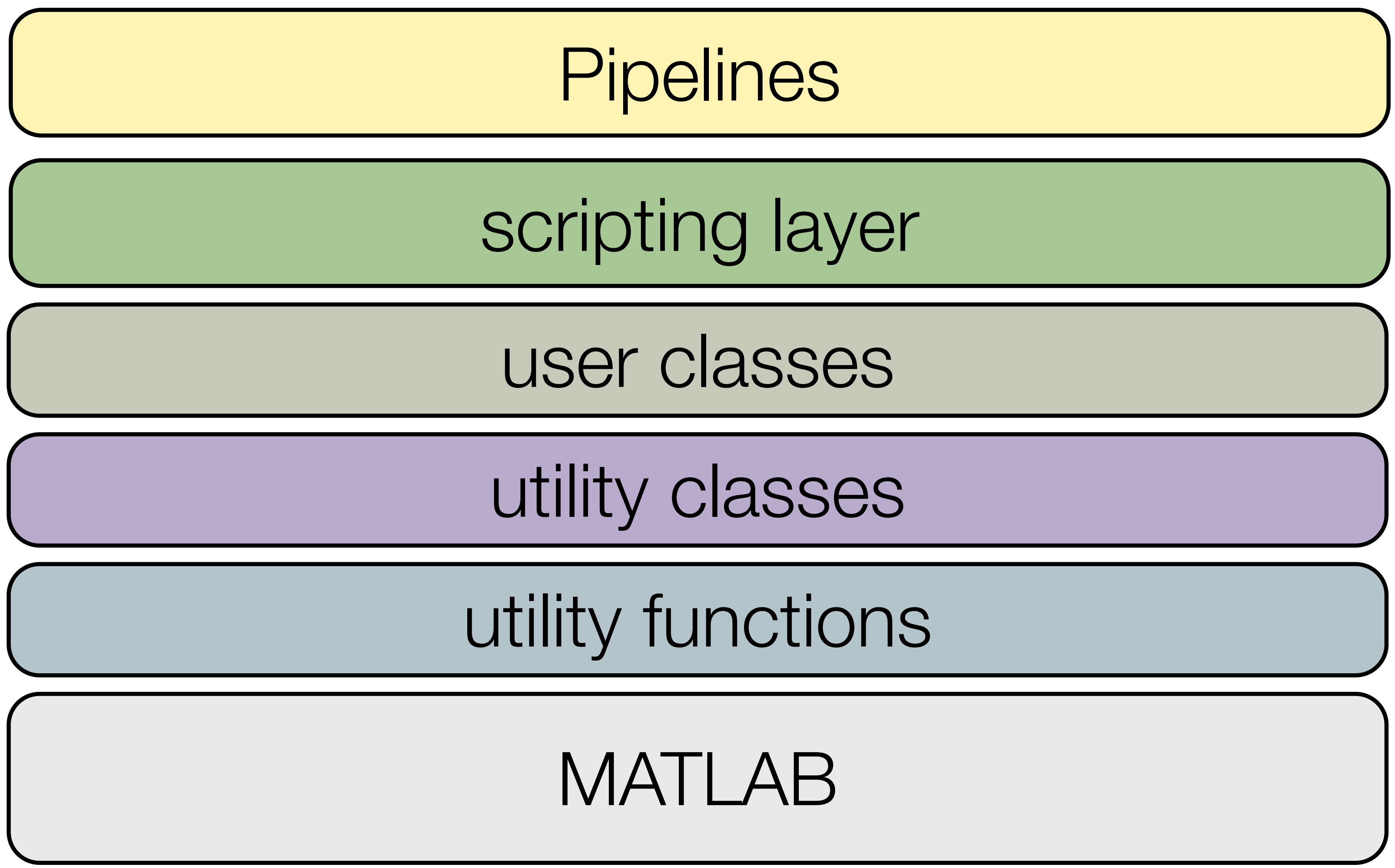
Pipelines

A pipeline represents a set of processing steps which, together, accomplish a particular aim



Pipelines

A pipeline represents a set of processing steps which, together, accomplish a particular aim



Pipelines

- We create pipelines designed for particular investigations:
  - quicklook of any investigation
  - conversion to acceleration (inv00001)
  - linear fit of x-axis sys id investigation (inv1001)
  - mcmc parameter estimation of x-axis sys id (inv1001)
  - ....
- The operational scripts then run these pipelines

```
%% Define investigation

inv00001_start = '2012-06-11 14:03:20';
inv00001_end   = '2012-06-11 19:56:40';

% Create investigation obj.

Inv00001 = Inv00001_001(inv00001_start,inv00001_end);

% build the pipeline for these
accPipe = STOCEstimate(Inv00001);
accPipe.get_data = false;


%%  Estimate acceleration

% from linear fit
accPipe.dataDirectory = 'linear_data';
linearParams = pest(fullfile(acc.dataDirectory,'fit_linear.mat'));
accPipe.estimate_acc(plist('params',linearParams));

% from mcmc
accPipe.dataDirectory = 'mcmc_data';
linearParams = pest(fullfile(acc.dataDirectory,'fit_mcmc.mat'));
accPipe.estimate_acc(plist('params',mcmcParams));
```

Interface with Mission Database

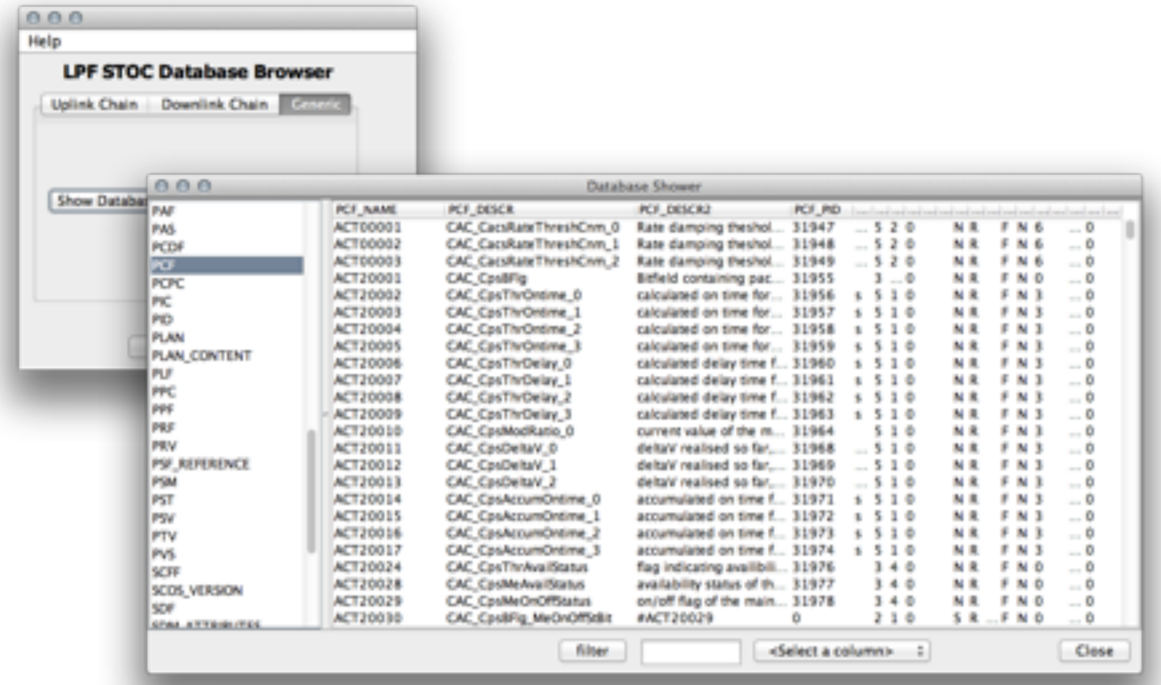
- STOC developed a mission database browser for use in operations
- We integrated this into LTPDA:
  - with a GUI
  - with command-line query tools

```
>> p = MIBrowser.getParamsWithIDs('GST50121')

p =

  tmdesc handle

  Properties:
      shortName: '5Hz_FEE_Tm1Pos2Z'
    description: 'SDM Filtered to 5Hz -
IS_FEE_Tm1Pos2Z'
             id: 'GST50121'
          units: [1x1 unit]
```



System Simulators

- Various simulators exist:
  - SVF (Software Verification Facility)
    - runs on-board software with software models of system
      - close to real-time
  - STOC Simulator
    - runs ‘real’ Payload Operation Requests (PORs) on an industry provided Simulink model of LPF
      - fairly detailed physical models, fast, complex
  - ESOC Simulator
    - runs on-board software, interfaces with main control system
      - realistic telemetry output
    - low-fidelity physical models, but functionally correct
  - Linear Statespace Models
    - integrated with DA software
    - simplified physical models compared to STOC Simulator
    - full 3D modular model of LPF
    - very fast and flexible

How we use the simulators

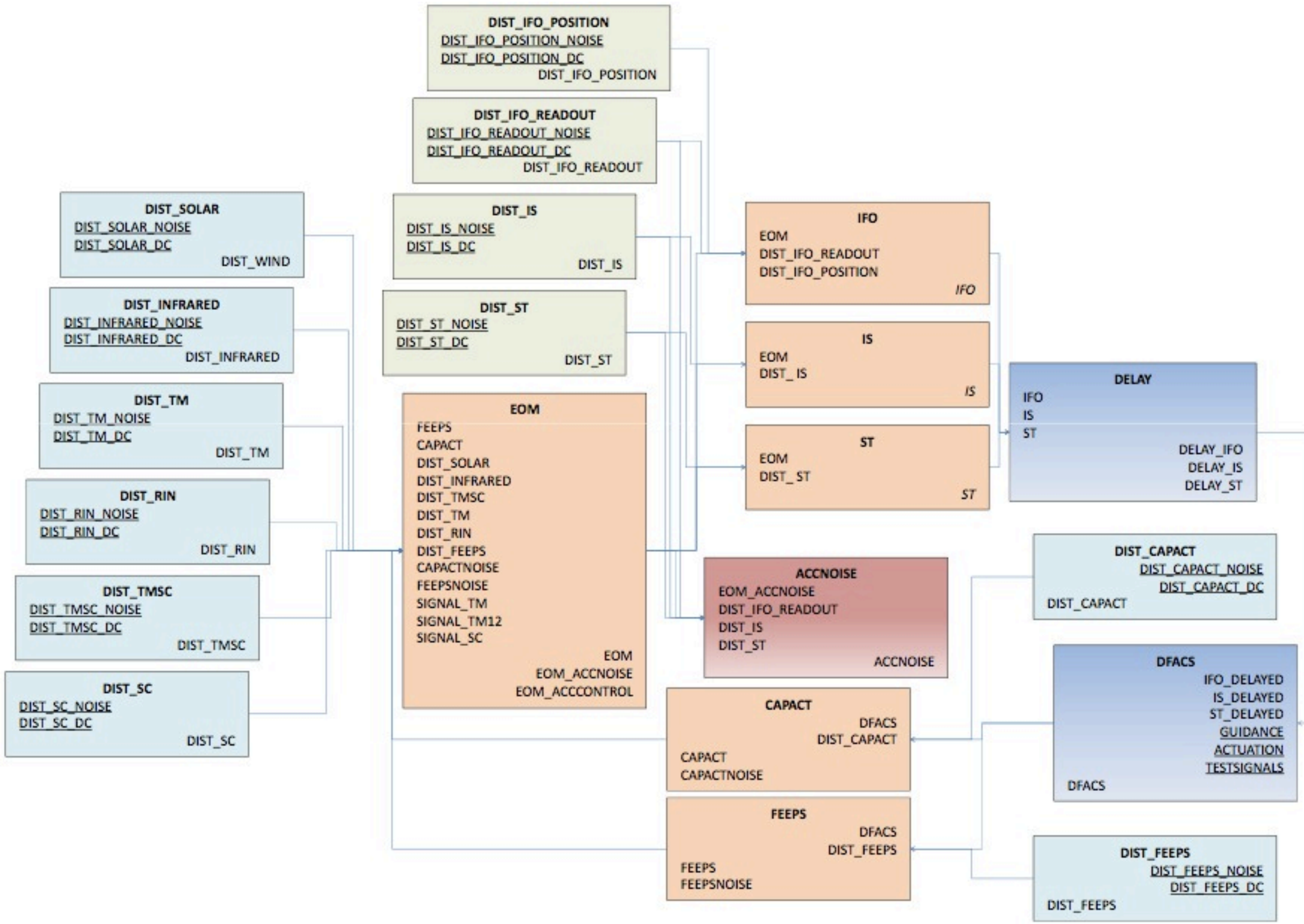
How we use the simulators

- STOC Simulator
  - validate PORs and investigations
  - generate data for STOC Simulations
    - contains more complexity than our DA statespace models
    - it is an ‘unknown’ system which we need to characterise

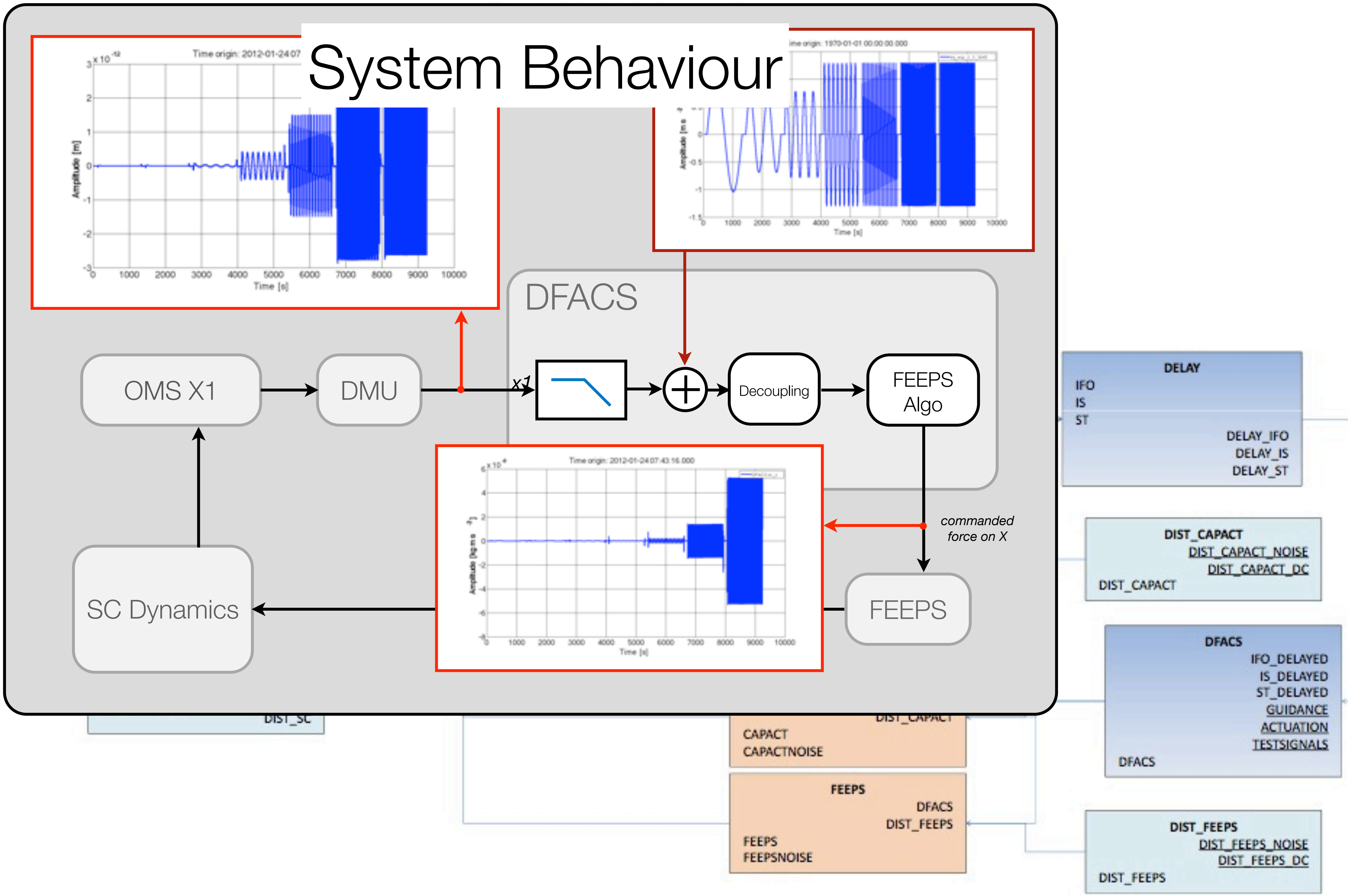
How we use the simulators

- STOC Simulator
  - validate PORs and investigations
  - generate data for STOC Simulations
    - contains more complexity than our DA statespace models
    - it is an ‘unknown’ system which we need to characterise
- DA Statespace Models
  - rapid prototyping of investigations
    - proof of concept in designing experiments
    - run time-domain simulations to produce test data sets
  - learning about system behaviour
    - what happens if I inject a signal here?
  - template generation for system identification
    - full 3D LPF model has >500 tunable parameters
    - fit particular parameter sets to outputs of particular experiments

LPF Model

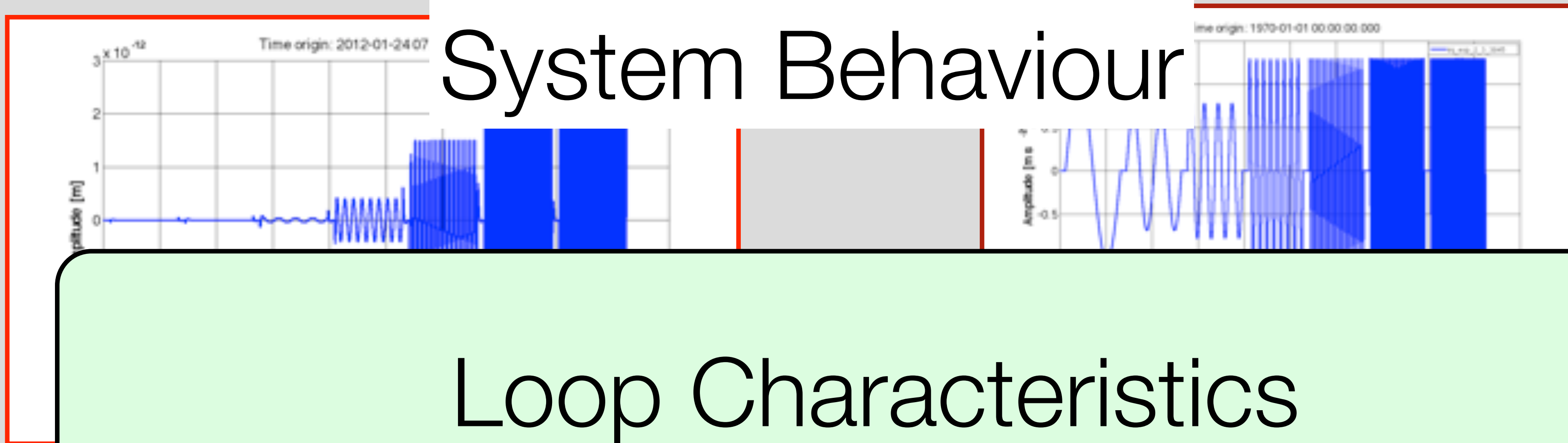

DIST_IFO_POSITION
DIST_IFO_POSITION_NOISE
DIST_IFO_POSITION_DC
DIST_IFO_POSITION
DIST_IFO_READOUT
DIST_IFO_READOUT_NOISE
DIST_IFO_READOUT_DC
DIST_IFO_READOUT
DIST_SOLAR
DIST_SOLAR_NOISE
DIST_SOLAR_DC
DIST_WIND
DIST_IS
DIST_IS_NOISE
DIST_IS_DC
DIST_IS
IFO
EOM
DIST_IFO_READOUT
DIST_IFO_POSITION
IFO
DIST_INFRARED
DIST_INFRARED_NOISE
DIST_INFRARED_DC
DIST_INFRARED
DIST_ST
DIST_ST_NOISE
DIST_ST_DC
DIST_ST
IS
EOM
DIST_ IS
IS
DELAY
IFO
IS
ST
DELAY_IFO
DELAY_IS
DELAY_ST
DIST_TM
DIST_TM_NOISE
DIST_TM_DC
DIST_TM
EOM
FEEPS
CAPACT
DIST_SOLAR
DIST_INFRARED
DIST_TMSC
DIST_TM
DIST_RIN
DIST_FEEPS
CAPACTNOISE
FEEPSNOISE
SIGNAL_TM
SIGNAL_TM12
SIGNAL_SC
EOM
EOM_ACCNOISE
EOM_ACCCONTROL
ST
EOM
DIST_ ST
ST
DIST_RIN
DIST_RIN_NOISE
DIST_RIN_DC
DIST_RIN
DIST_CAPACT
DIST_CAPACT_NOISE
DIST_CAPACT_DC
DIST_CAPACT
ACCNOISE
EOM_ACCNOISE
DIST_IFO_READOUT
DIST_IS
DIST_ST
ACCNOISE
DIST_TMSC
DIST_TMSC_NOISE
DIST_TMSC_DC
DIST_TMSC
DIST_SC
DIST_SC_NOISE
DIST_SC_DC
DIST_SC
CAPACT
DFACS
DIST_CAPACT
CAPACT
CAPACTNOISE
DFACS
IFO_DELAYED
IS_DELAYED
ST_DELAYED
GUIDANCE
ACTUATION
TESTSIGNALS
DFACS
FEEPS
DFACS
DIST_FEEPS
FEEPS
FEEPSNOISE
DIST_FEEPS
DIST_FEEPS_NOISE
DIST_FEEPS_DC
DIST_FEEPS

LPF Model

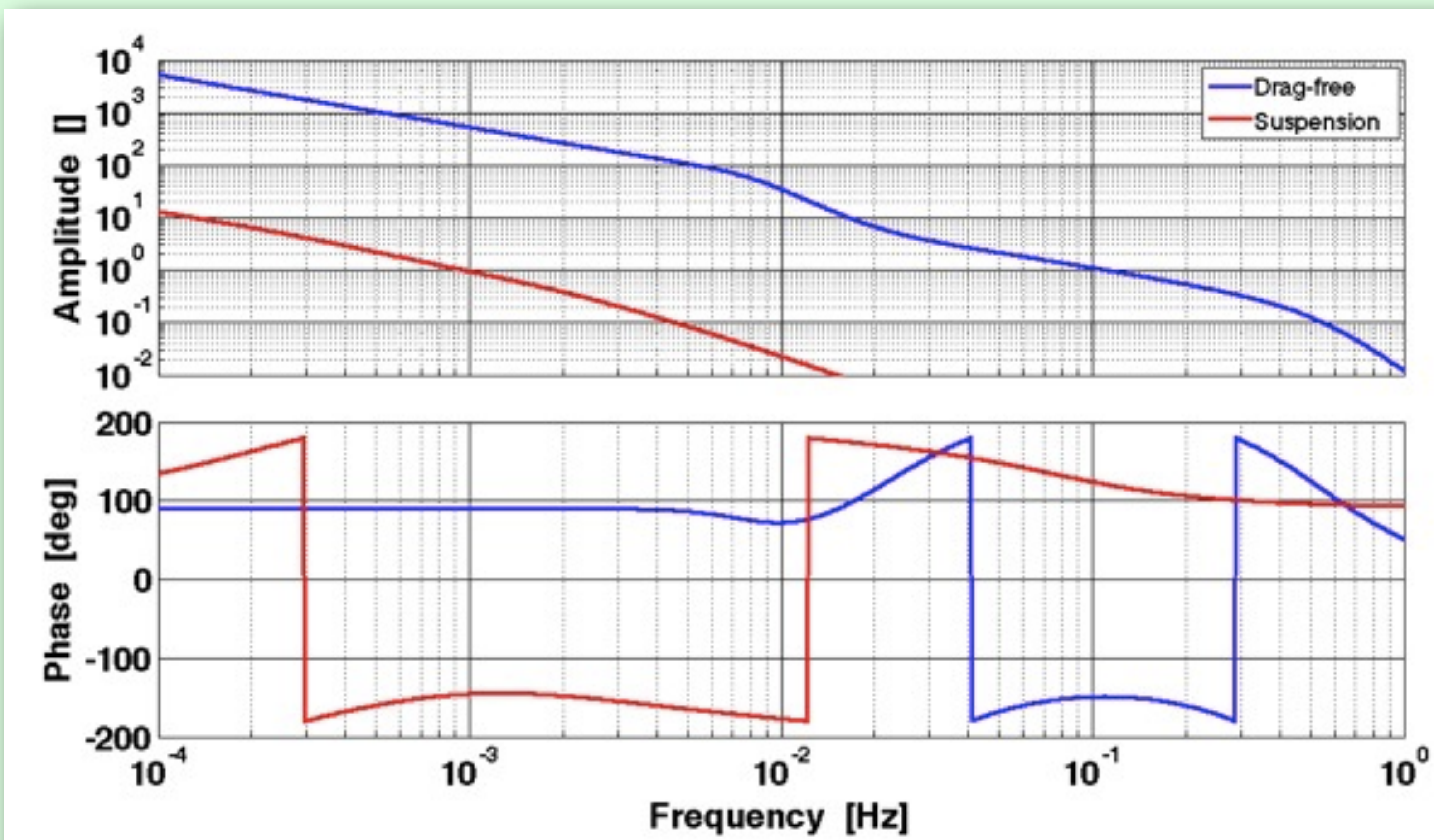

System Behaviour
DFACS
OMS X1
DMU
x1
Decoupling
FEEPS Algo
commanded force on X
FEEPS
SC Dynamics
Time [s]
DELAY
IFO
IS
ST
DELAY_IFO
DELAY_IS
DELAY_ST
DIST_CAPACT
DIST_CAPACT_NOISE
DIST_CAPACT_DC
DIST_CAPACT
DFACS
IFO_DELAYED
IS_DELAYED
ST_DELAYED
GUIDANCE
ACTUATION
TESTSIGNALS
DFACS
DIST_SC
CAPACT
CAPACTNOISE
FEEPS
DFACS
DIST_FEEPS
FEEPS
FEEPSNOISE
DIST_FEEPS
DIST_FEEPS_NOISE
DIST_FEEPS_DC
DIST_FEEPS

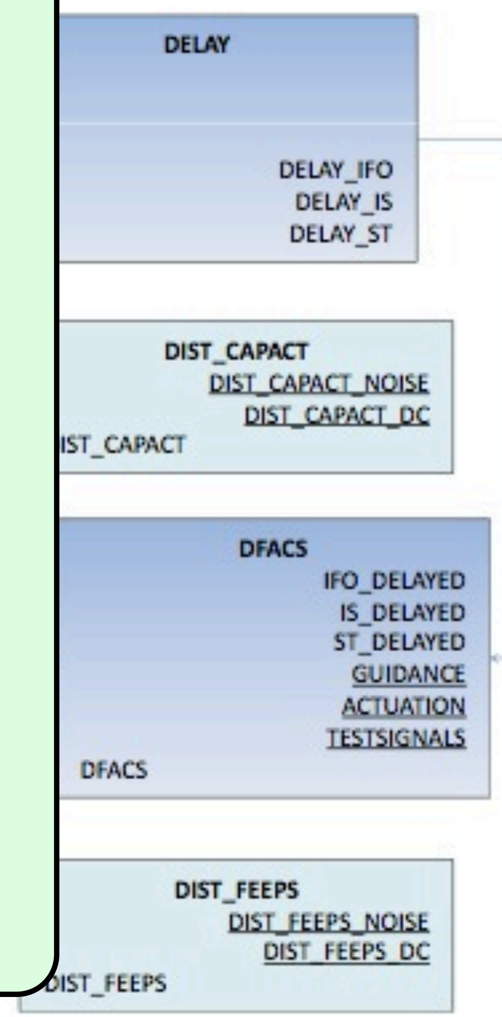
LPF Model

System Behaviour

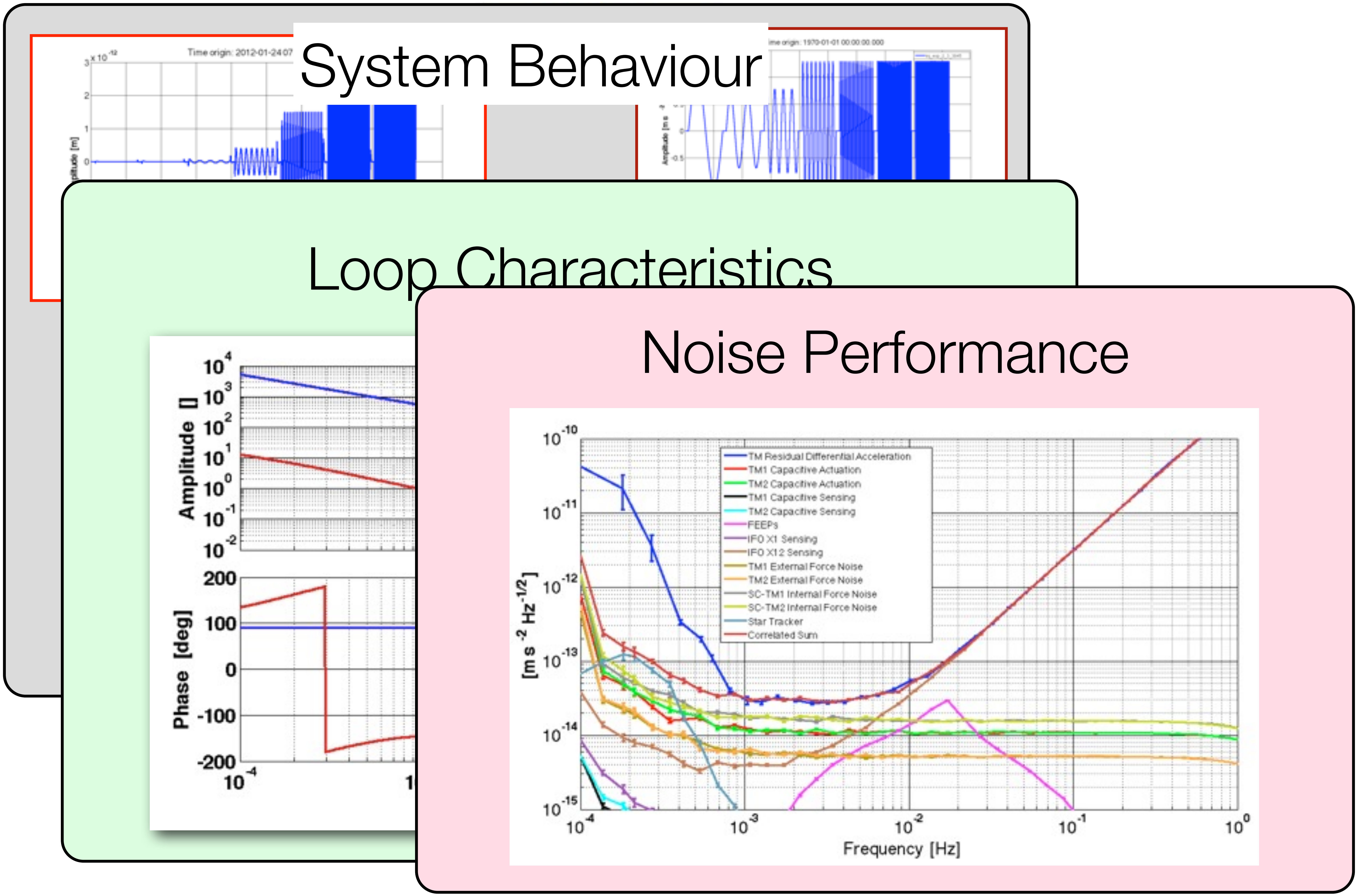


Loop Characteristics



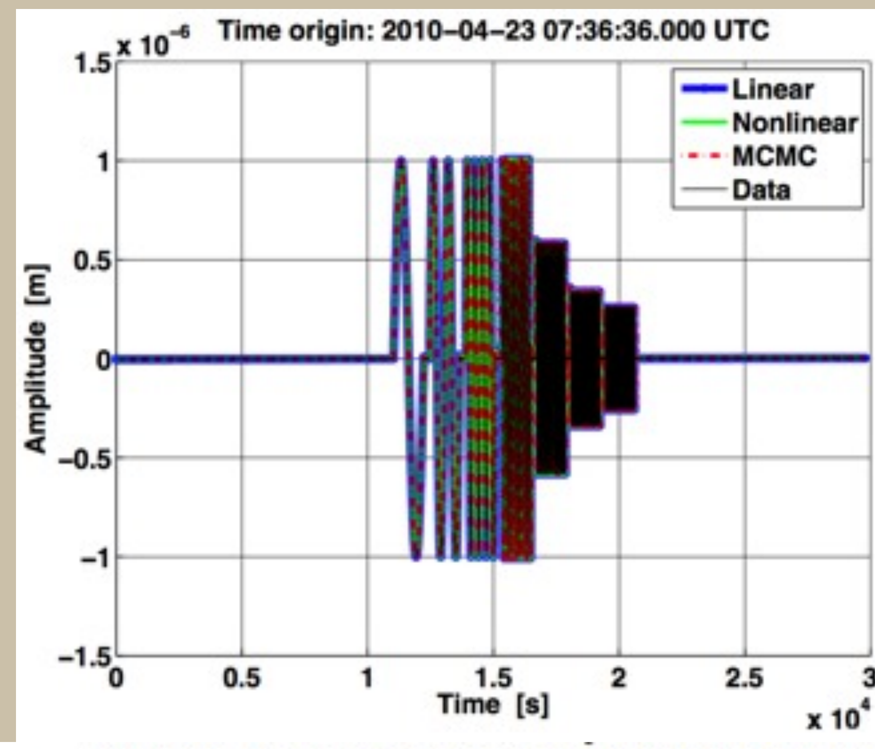


LPF Model


System Behaviour
Loop Characteristics
Amplitude []
Phase [deg]
Noise Performance
TM Residual Differential Acceleration
TM1 Capacitive Actuation
TM2 Capacitive Actuation
TM1 Capacitive Sensing
TM2 Capacitive Sensing
FEEPs
IFO X1 Sensing
IFO X12 Sensing
TM1 External Force Noise
TM2 External Force Noise
SC-TM1 Internal Force Noise
SC-TM2 Internal Force Noise
Star Tracker
Correlated Sum
[m s -2 Hz -1/2]
Frequency [Hz]

LPF Model

System Identification



Parameter	Linear $\hat{x} \pm \sigma$	Non-linear $\hat{x} \pm \sigma \, (\sigma/\sigma_{CR})$	MCMC $\hat{x} \pm \sigma \, (\sigma/\sigma_{CR})$
A1	1.0699 ± 0.0005	1.0705 ± 0.0006	1.0701 ± 0.0003
A2	0.99998 ± 0.00003	0.99998 ± 0.00003	0.99997 ± 0.00002
S21	$(1.2 \pm 0.4) \times 10^{-6}$	$(1.2 \pm 0.4) \times 10^{-6}$	$(2.0 \pm 0.2) \times 10^{-6}$
del1	-0.1982 ± 0.0005	-0.1985 ± 0.0005	-0.2020 ± 0.0001
del2	-0.199 ± 0.001	-0.199 ± 0.001	-0.1995 ± 0.0008
ω_2^1	$(-1.319 \pm 0.002) \times 10^{-6}$	$(-1.319 \pm 0.002) \times 10^{-6}$	$(-1.320 \pm 0.001) \times 10^{-6}$
$\omega_2^2 - \omega_1^2$	$(-7.160 \pm 0.006) \times 10^{-7}$	$(-7.160 \pm 0.006) \times 10^{-7}$	$(-7.148 \pm 0.006) \times 10^{-7}$

Experiments and investigations

- We break the mission into experiment phases
  - noise hunting, system identification, environmental characterisation, etc
  - each experiment phase is broken down into investigations
- The mission time-line is built up from these elements
- The time-line can be adjusted to
  - rearrange planned investigations
  - include new investigations
    - needs to be done a few days in advance

Process of defining experiments

Process of defining experiments

- Two main strands of definition:
  - hardware providers developed targeted strategies
    - OMS characterisation experiments, Radiation monitor activities, ...
  - system level characterisation coming from DA team
    - determine system parameters for physical model
      - gains, stiffnesses, cross-couplings, etc

Process of defining experiments

- Two main strands of definition:
  - hardware providers developed targeted strategies
    - OMS characterisation experiments, Radiation monitor activities, ...
  - system level characterisation coming from DA team
    - determine system parameters for physical model
      - gains, stiffnesses, cross-couplings, etc
- The scheme has been to write a technical note which:
  - describes possible experiments
  - discusses the aim of those experiments
  - details how they might be analysed and what the expected results are
  - defines the telemetry needed

Process of defining experiments (II)

Process of defining experiments (II)

- STOC takes these proposals and defines investigations
  - validation involves:
    - confirming the experiment is feasible
    - defining telemetry sets
    - simulating and testing

Process of defining experiments (II)

- STOC takes these proposals and defines investigations
  - validation involves:
    - confirming the experiment is feasible
    - defining telemetry sets
    - simulating and testing
- These are then used to develop the mission time-line with the overall aim of:
  - achieving the desired level of free-fall
  - developing a detailed physical model of the system

Aims of experiment phases

Aims of experiment phases

- system identification needed to:
  - tune the instrument for best performance
  - understand new behaviour
  - subtract noise contributions

Aims of experiment phases

- system identification needed to:
  - tune the instrument for best performance
  - understand new behaviour
  - subtract noise contributions
- characterisation of glitches, noise non-stationarity, data quality
  - input to subsequent astrophysical data analysis

Aims of experiment phases

- system identification needed to:
  - tune the instrument for best performance
  - understand new behaviour
  - subtract noise contributions
- characterisation of glitches, noise non-stationarity, data quality
  - input to subsequent astrophysical data analysis
- characterisation of environment
  - temperature and magnetic effects
  - may affect data quality, may require tuning the system to minimise couplings

How does all this fit to LISA?

How does all this fit to LISA?

- Commissioning of LISA
  - full LPF mission is a commissioning/characterisation exercise

How does all this fit to LISA?

- Commissioning of LISA
  - full LPF mission is a commissioning/characterisation exercise
- Continuous maintenance and characterisation
  - system identification to maintain peak performance
  - characterisation of glitches, noise non-stationarity, data quality
    - input to subsequent astrophysical data analysis

How does all this fit to LISA?

- Commissioning of LISA
  - full LPF mission is a commissioning/characterisation exercise
- Continuous maintenance and characterisation
  - system identification to maintain peak performance
  - characterisation of glitches, noise non-stationarity, data quality
    - input to subsequent astrophysical data analysis
- LISA activities will need similar tools and experiments
  - are we missing experiments on LPF which may provide useful input to LISA development?

How does all this fit to LISA?

- Commissioning of LISA
  - full LPF mission is a commissioning/characterisation exercise
- Continuous maintenance and characterisation
  - system identification to maintain peak performance
  - characterisation of glitches, noise non-stationarity, data quality
    - input to subsequent astrophysical data analysis
- LISA activities will need similar tools and experiments
  - are we missing experiments on LPF which may provide useful input to LISA development?
- LPF provides real data under a real operational scenario for a 'similar' instrument

Infrastructure

Infrastructure

- Data handling
  - learn from the short-comings of the LPF system and improve from LISA
    - parameter naming, time stamping, data delivery system, etc

Infrastructure

- Data handling
  - learn from the short-comings of the LPF system and improve from LISA
    - parameter naming, time stamping, data delivery system, etc
- Data analysis (commissioning)
  - take concepts and algorithms from LTPDA and design a tool suitable for LISA
    - maybe LTPDA would do, maybe we want to move away from MATLAB
    - but keep: history tracking, object oriented, testing, dynamic documentation, etc

Infrastructure

- Data handling
  - learn from the short-comings of the LPF system and improve from LISA
    - parameter naming, time stamping, data delivery system, etc
- Data analysis (commissioning)
  - take concepts and algorithms from LTPDA and design a tool suitable for LISA
    - maybe LTPDA would do, maybe we want to move away from MATLAB
    - but keep: history tracking, object oriented, testing, dynamic documentation, etc
- Documentation
  - tools to do ‘live’ documenting (logging) of DA activity are few and far between
    - wikis, local wordprocessor/tex files, other e-labbooks exists
      - all are poor or slow
  - write requirements and start thinking about a design for a tool which fits

Infrastructure

- Data handling
  - learn from the short-comings of the LPF system and improve from LISA
    - parameter naming, time stamping, data delivery system, etc
- Data analysis (commissioning)
  - take concepts and algorithms from LTPDA and design a tool suitable for LISA
    - maybe LTPDA would do, maybe we want to move away from MATLAB
    - but keep: history tracking, object oriented, testing, dynamic documentation, etc
- Documentation
  - tools to do 'live' documenting (logging) of DA activity are few and far between
    - wikis, local wordprocessor/tex files, other e-labbooks exists
      - all are poor or slow
  - write requirements and start thinking about a design for a tool which fits
- Simulators
  - the LPF simulators have proved invaluable
    - modelling, learning, simulating, parameter identification, etc
  - take modular concepts of the LPF statespace simulator and design a LISA system simulator
    - bottom up
    - capture behaviour, not all details

Experiments

- Test/develop characterisation experiments for LISA
  - base these on those designed for LPF
  - we need a system simulator for this

- Other experiments will be needed
  - what are these?
  - can they be somehow tested on LPF?

- These experiments can help drive/develop requirements on the instrument(s)

What else do we get from LPF?

- operations experience
  - both at ESA and in LPF/LISA science community
  - try to keep the science team from LPF in the LISA project
  - ensure we write detailed reports on LPF observed behaviour (planned, but hard)
- first set of data indicating how this type of instrument will behave
  - hardware characteristics (glitches, noise levels, failure modes, long-term performance)
  - system behaviour (instabilities, tuning performance, robustness)
  - long-term behaviour
- Look for similar behaviour when developing and integrating LISA

Lessons to learn from preparing LPF Ops

Lessons to learn from preparing LPF Ops

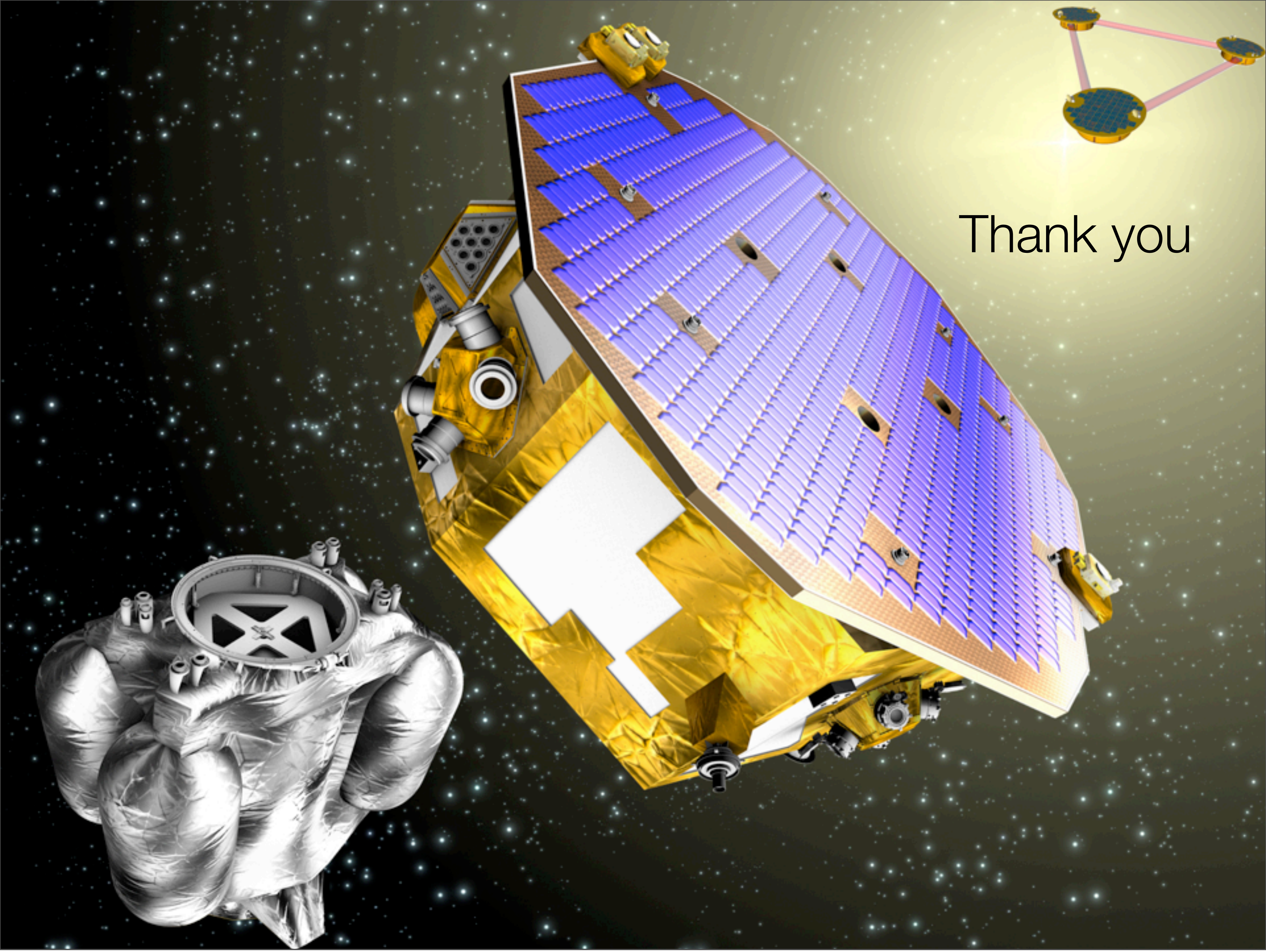
- Instrument user manuals are not enough!
- Have close interaction of Ops/DA team with hardware providers
  - do this early in the development to avoid 'surprises' when preparing operations
  - development of DA was probably too late in LPF
    - it was often too late to impose hardware changes
      - lots of work-arounds and compromises

Lessons to learn from preparing LPF Ops

- Instrument user manuals are not enough!
- Have close interaction of Ops/DA team with hardware providers
  - do this early in the development to avoid ‘surprises’ when preparing operations
  - development of DA was probably too late in LPF
    - it was often too late to impose hardware changes
      - lots of work-arounds and compromises
- Include Ops/DA team in hardware test campaigns
  - provides invaluable experience to the Ops/DA team
    - how to operate instruments, experience which is not typically captured in user manuals
  - gives valuable feedback to hardware providers
    - do this early to avoid the ‘we don’t have time to worry about that now’ problem

Lessons to learn from preparing LPF Ops

- Instrument user manuals are not enough!
- Have close interaction of Ops/DA team with hardware providers
  - do this early in the development to avoid ‘surprises’ when preparing operations
  - development of DA was probably too late in LPF
    - it was often too late to impose hardware changes
      - lots of work-arounds and compromises
- Include Ops/DA team in hardware test campaigns
  - provides invaluable experience to the Ops/DA team
    - how to operate instruments, experience which is not typically captured in user manuals
  - gives valuable feedback to hardware providers
    - do this early to avoid the ‘we don’t have time to worry about that now’ problem
- People leave: it’s important to keep a body of knowledge alive from hardware construction/testing through to Ops
  - integrating a wider range of people in the development process can help with this


Thank you