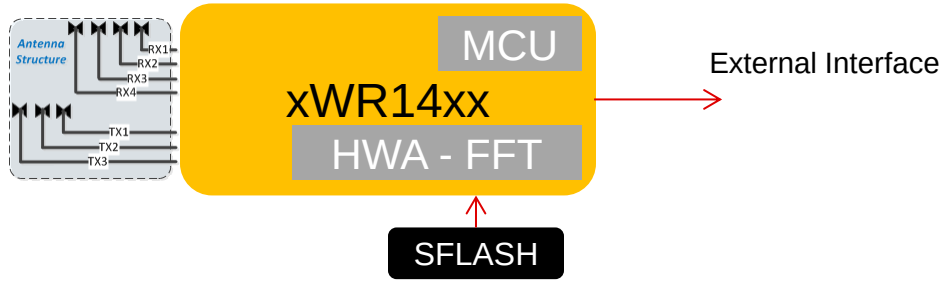
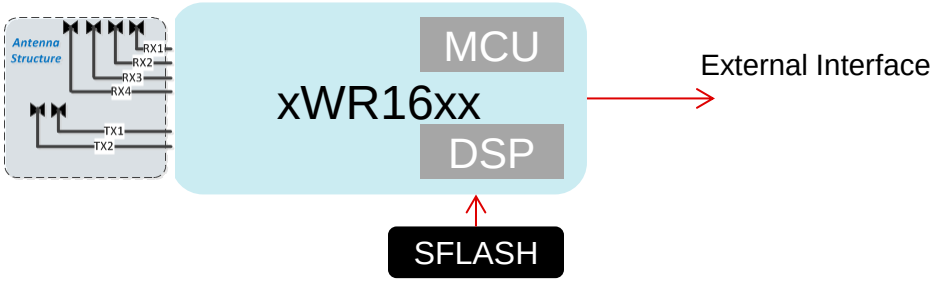
mmWave SDK

(Software offering for Single Chip TI mmWave RADARs)

Thu 5/11/2017

Software Offering - Single Chip TI mmWave RADARs





Platform Software

- RTOS
- Device Drivers
- APIs and Processing Libraries
- Sample Applications
- Tools

Software Offering → mmWave Software Development Kit

TI mmWave SDK – Sensing & Analytics SW Suite

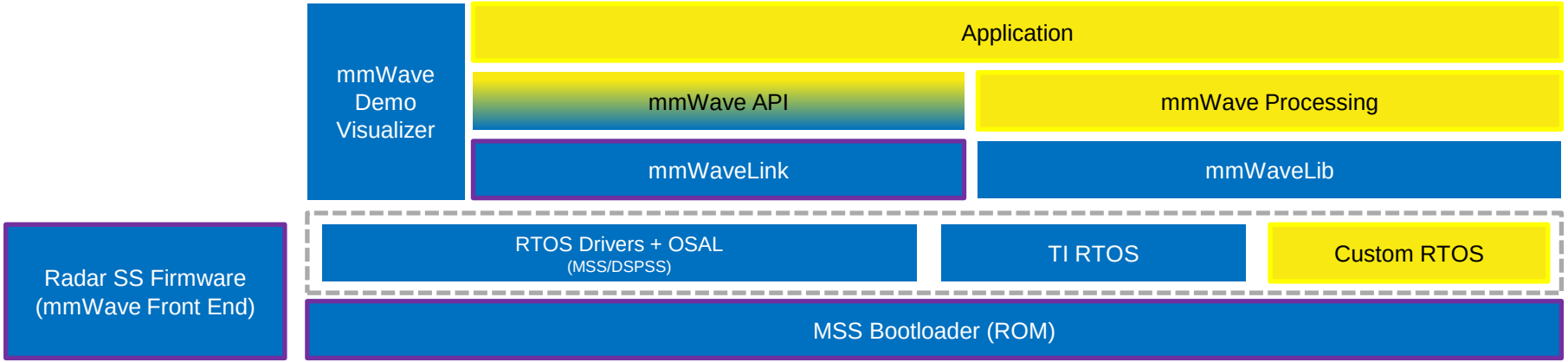
TI code

Customer code

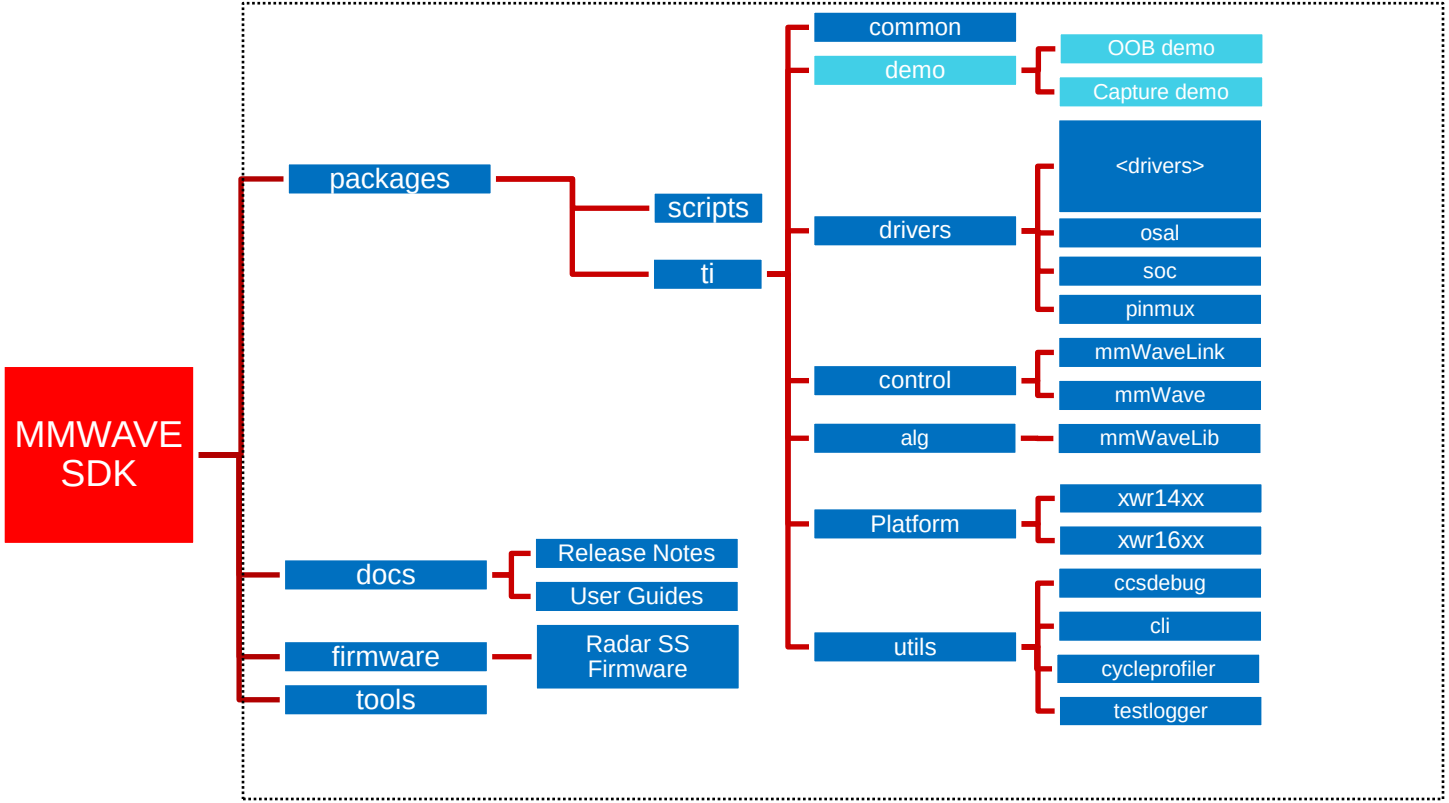
ISO 26262 enabled code

Production Quality TI Code

- Modular design
- Well defined APIs
- Documentation – doxygen, release notes, user guide
- Code coverage analysis



mmWave SDK - Packaging



- Uses TI CGT compiler (R4F, C674X) provided as part of CCS
- Demo built over TI RTOS
- Simple makefile based build system

DRIVERS

API

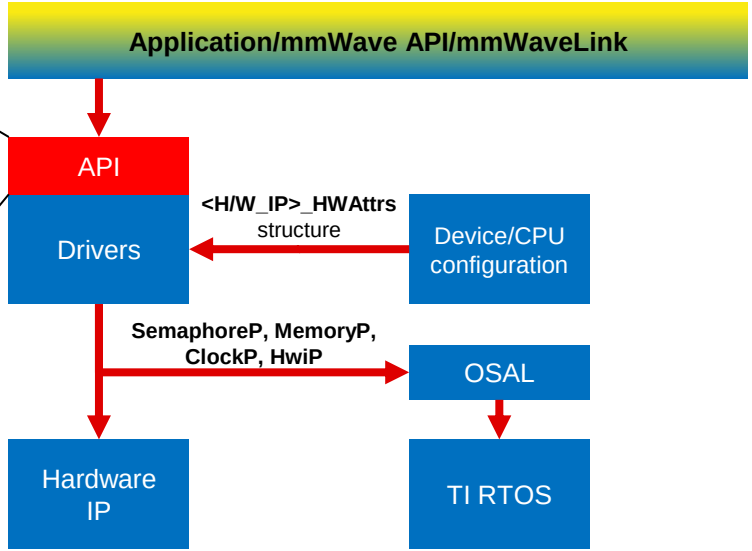
void **ADCBuf_init**(void);

void **ADCBuf_Params_init**(ADCBuf_Params *params);

ADCBuf_Handle **ADCBuf_open**(uint8_t index, ADCBuf_Params *params);

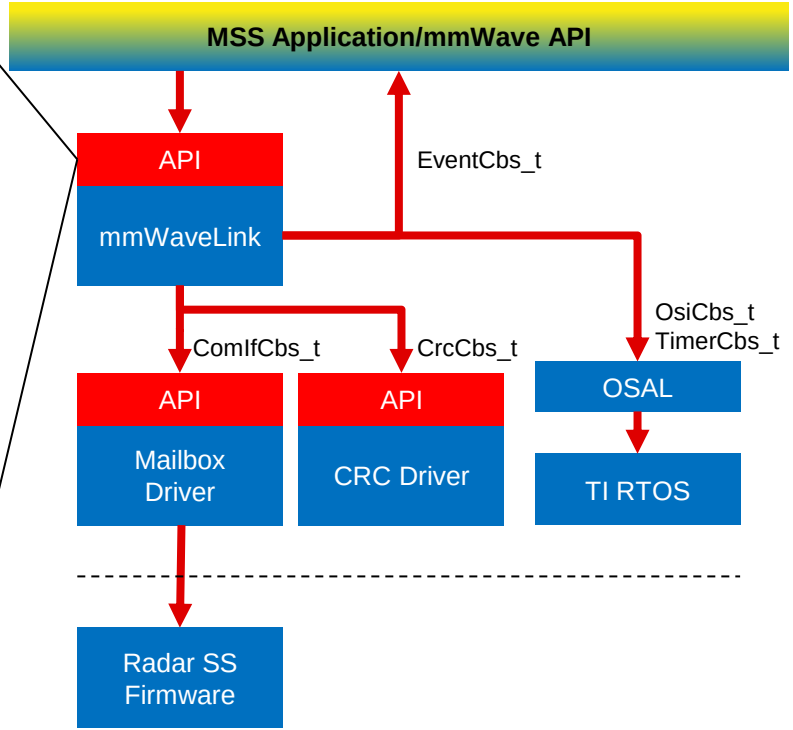
int16_t **ADCBuf_control**(ADCBuf_Handle handle, uint8_t cmd, void * arg);

void **ADCBuf_close**(ADCBuf_Handle handle);

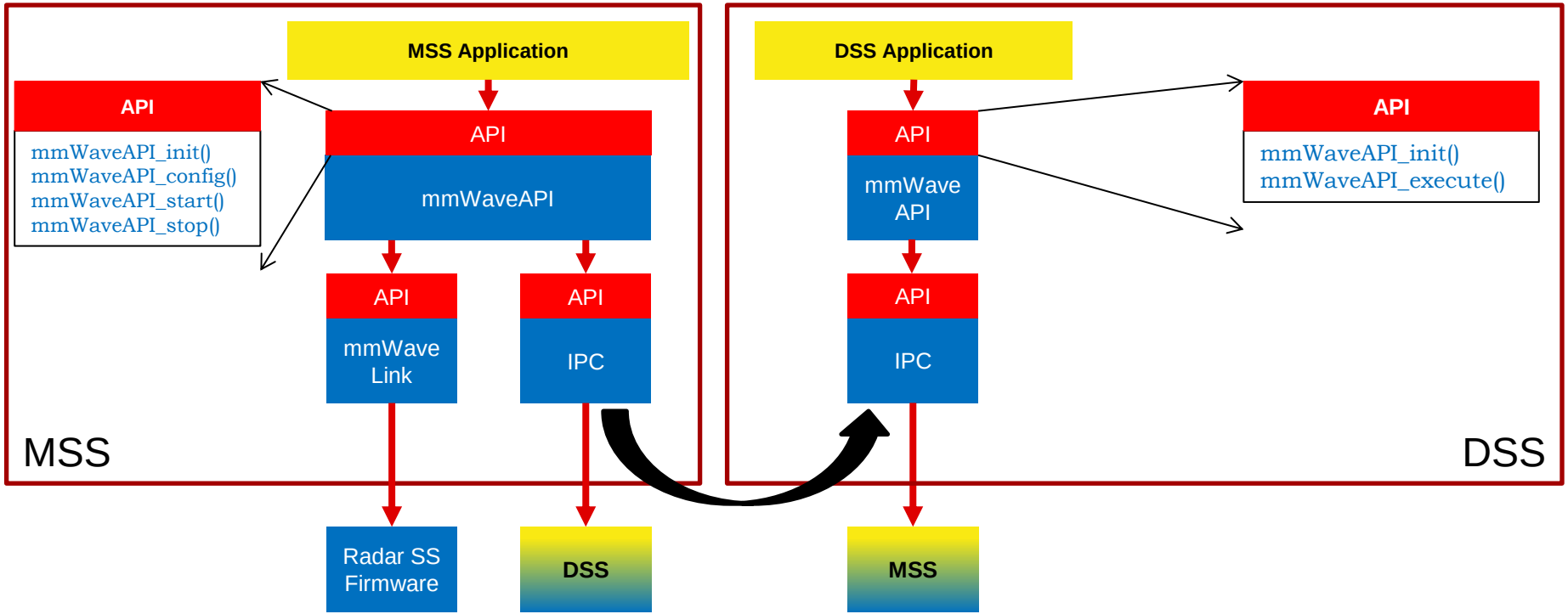


mmWaveLink

APIs	Description
Device Manager APIs	
rlDevicePowerOn	Initializes the driver and handshake with mmWave Front End
rlDevicePowerOff	De-initializes the driver and stops the mmWave front end
Sensor Control APIs	
rlSetChannelConfig	Configures the number of RX and TX channels
rlSetAdcOutConfig	Configures the ADC format (# bits, Real/Complex)
rlSetProfileConfig	Configures the profile (Frequency start, Frequency slope, Idle time, RX gain, ADC sampling rate, HPF1 and HPF2 cutoff, TX output power, TX phase shifter)
rlSetChirpConfig	Configures variable part for frequency start, slope, ADC start time, idle time and selection of TX for each chirp
rlSetFrameConfig	Configures the frame (start chirp index, end chirp index, number of chirp loops, frame periodicity)
rlSensorStart	Triggers the transmission of the frames as per the frame and chirp configuration
rlSensorStop	Stops the transmission of the frames.

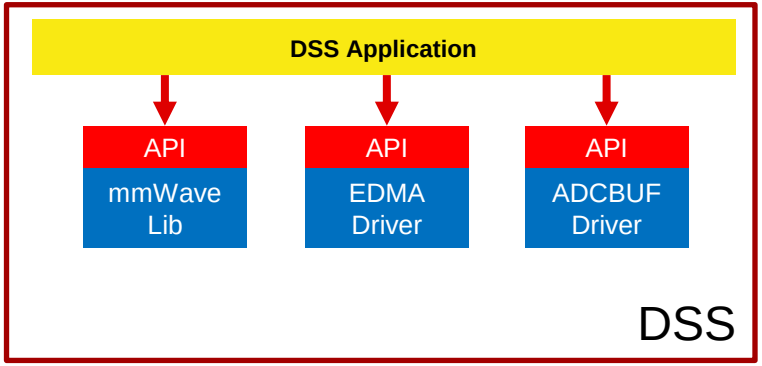


mmWave: Higher Level Control API

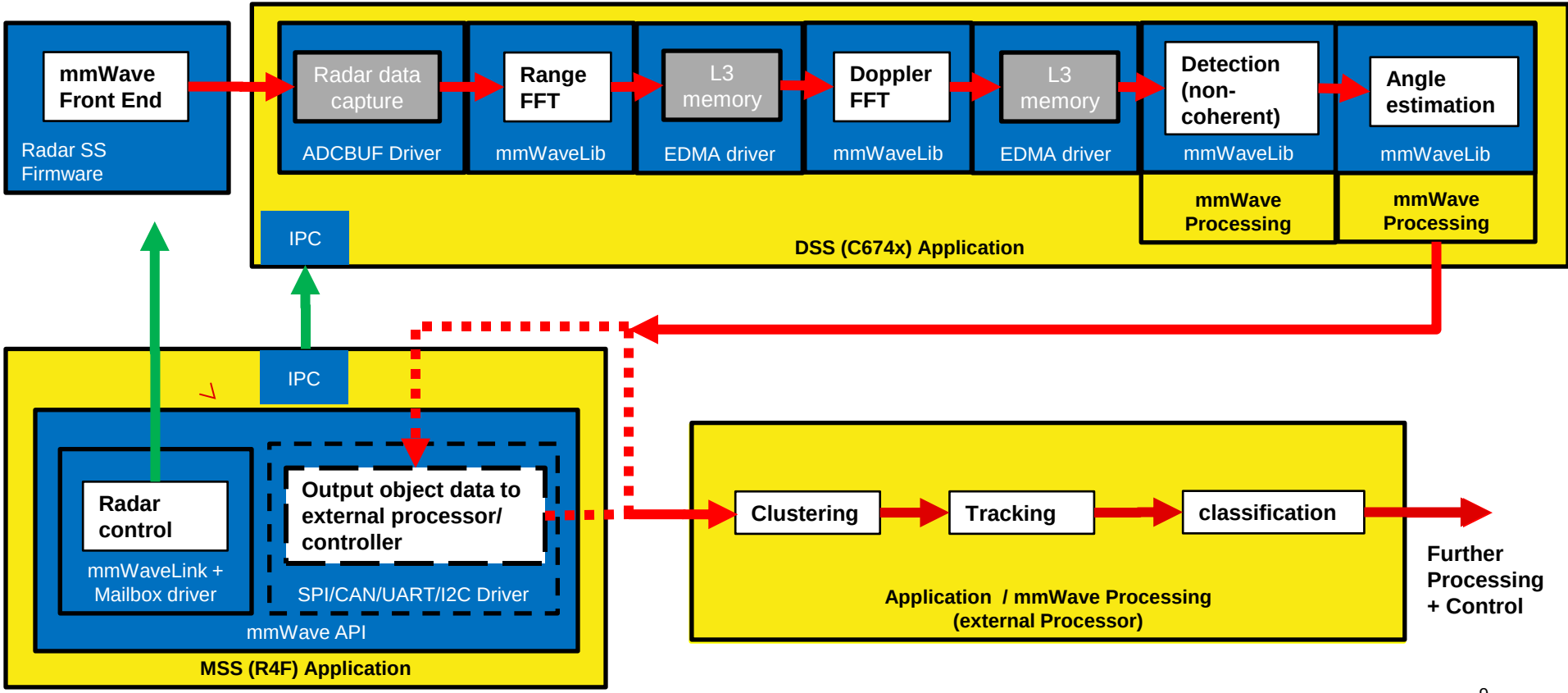


mmWaveLib

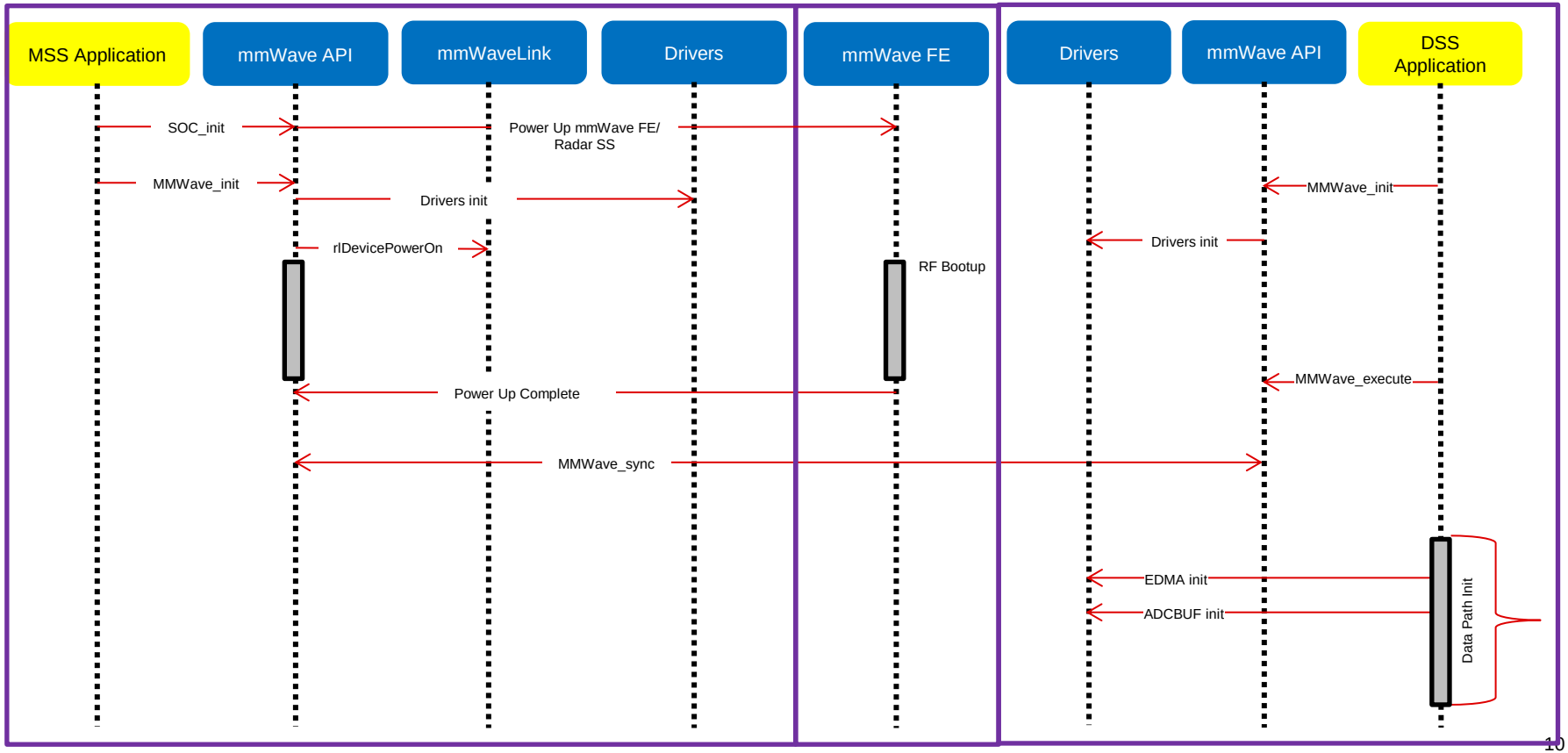
- Key routines required for FMCW radar-cube processing
  - FFT
    - 16 bit input and output
    - 16 bit input and 32 bit output
    - 32 bit input and 32 bit output
    - Single precision floating point variations provided as well
  - Block DFT – 16 point
  - Windowing
    - 16 bit input and output
    - 16 bit input and 32 bit output
  - Detection
    - CFAR-CA Detector
  - Log2Abs
    - Log2 of absolute value of a 32 bit complex number
  - Angle estimation
    - 16 point FFT with zero padded input (Angle FFT)
- Other Helper routines (Scaling, Shifting, Accumulation)
- Routines are optimized for C674x architecture



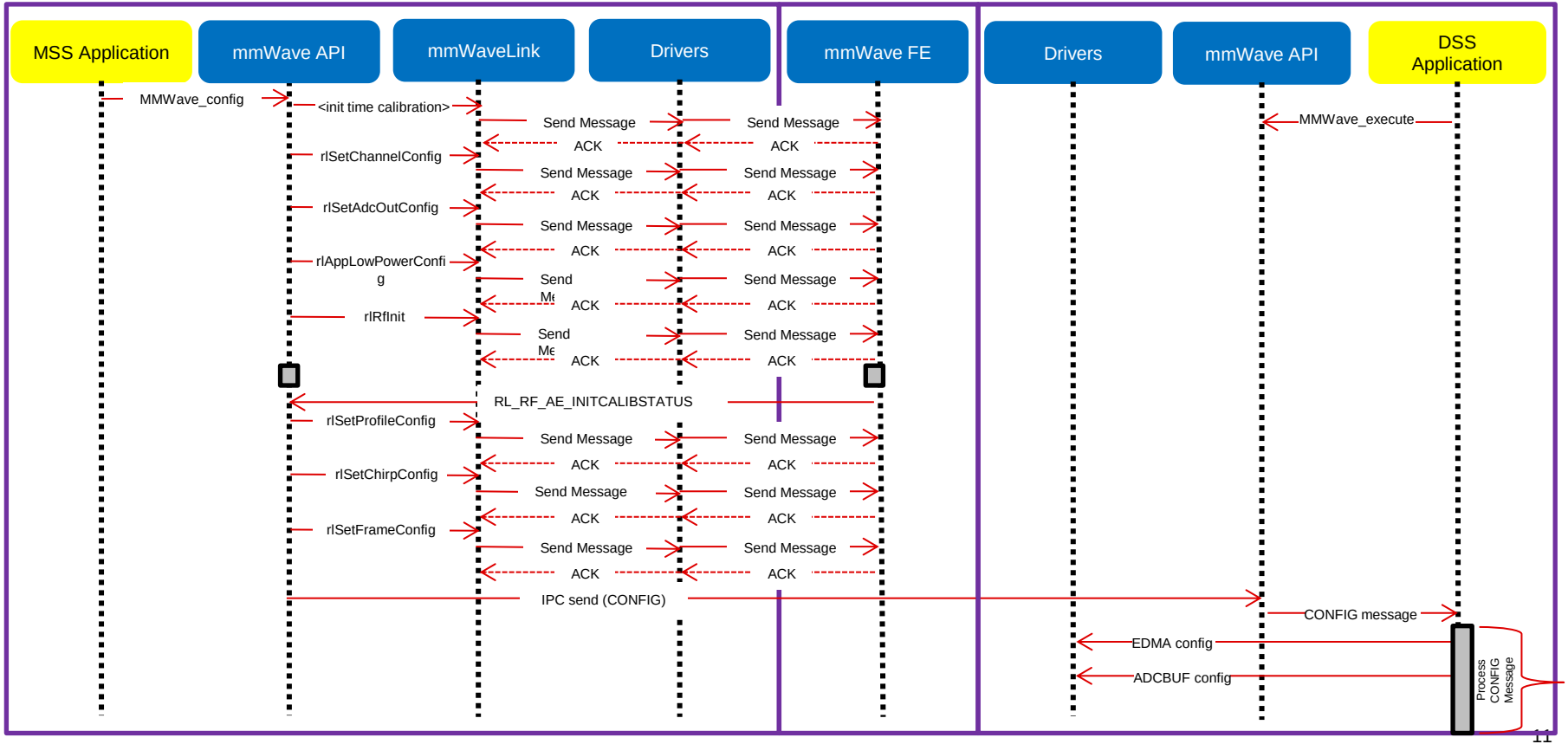
xWR1642 Processing Chain using mmWave SDK



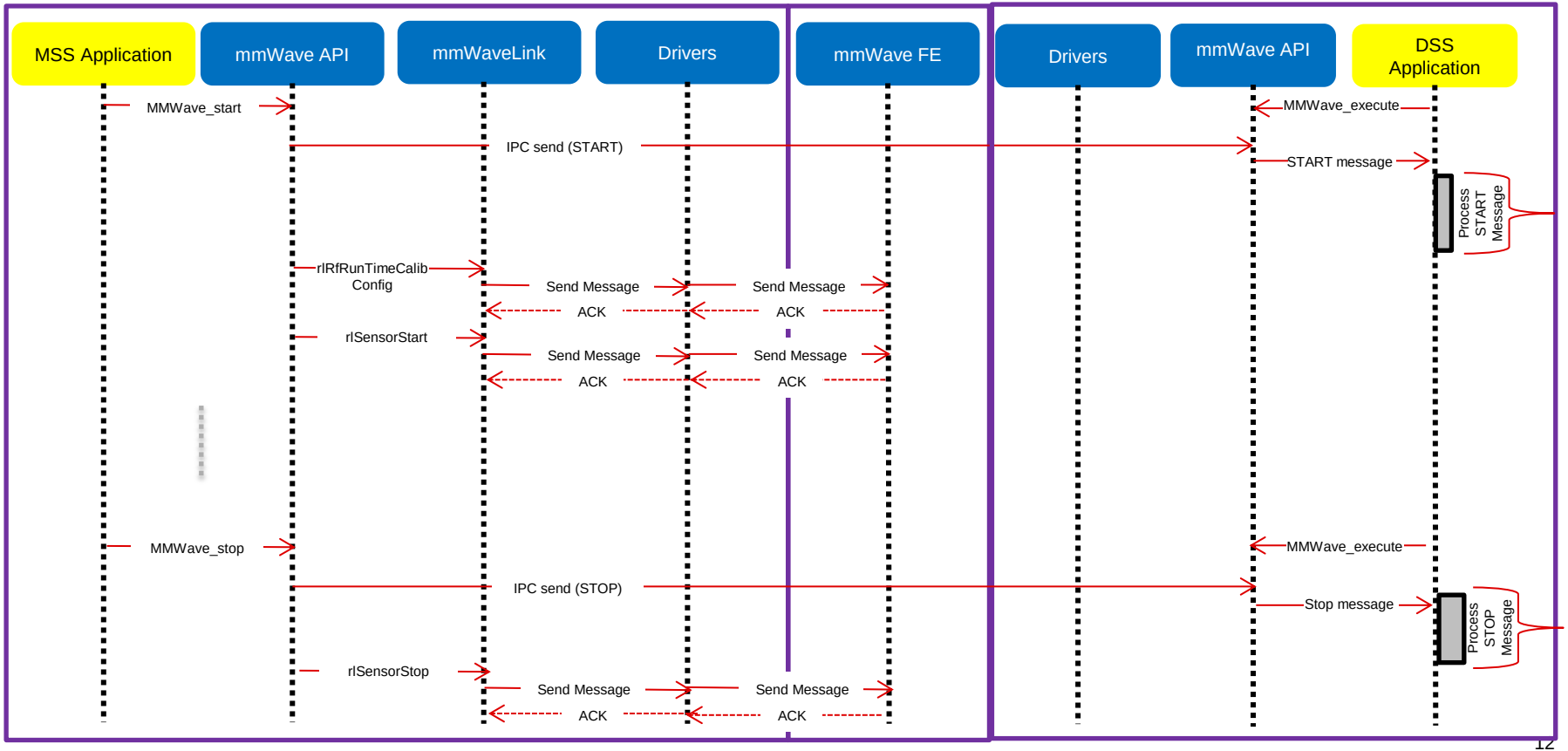
xWR1642: Detailed Control Flow (1/3)



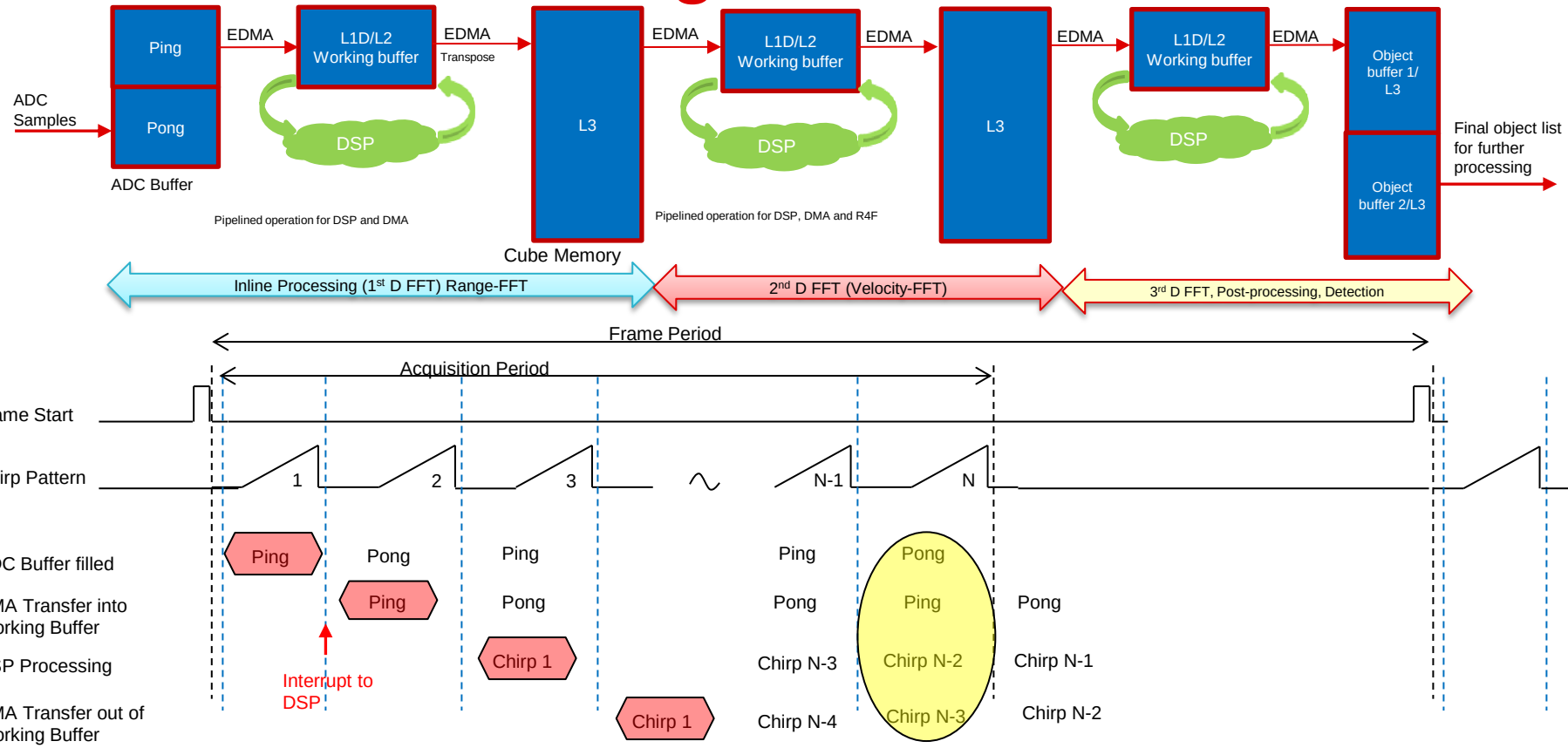
xWR1642: Detailed Control Flow (2/3)



xWR1642: Detailed Control Flow (3/3)



Detailed Processing Data Flow



Thank You