Heterogeneous SoC project

The vision is to have an easy to replicate, expandable demo that shows OpenAMP/LITE technologies running on a board or a simulator and that can easily be ported to other platforms.

The demo will showcase when different cores are used for different purposes, such as:

- Cortex A core for a general compute/connectivity tasks running Linux
- Cortex R/M running specialized tasks running and RTOS (Zephyr), including
    - Real-time tasks
    - Tasks that needs to be certified for functional safety
    - General offload

Boards:

- Xilinx Ultra96 board that contains 4xA53 + 2xR5
- ST stm32mp1 board that contains 2xA7 + 1xM4
- QEMU simulator for specific boards, preferably upstream QEMU
- Other boards from other vendors to be added with different Cortex As and Cortex Rs/Ms

Software stack:

- Linux running on the Cortex A connected to the outside world (terminal window, internet access, etc.)
- Zephyr running on the Cortex R/M connected to a simulated/fake device that measures something, say temperatures
    - Other RTOSes and bare metal to be added
- Linux application that instantiates the Zephyr applications and provides file and socket access and sends commands to the Zephyr app
- Zephyr application(s) that listens to commands from the Linux side and performs things like
    - Gets the current time from an internet site over a socket connection
    - Logs the fake temperature value and a timestamp to a file on the Linux side
    - Dumps a large memory area to Linux

Demo to show case (initial ideas):

- Configuration
    - Use System Device Tree to configure the complete system
        - Including which device goes where, how much physical memory to allocate to each domain, virtio channels/shared memory
        - Lopper would create traditional device tree for Linux
- Lifecycle
    - Linux app to use remoteproc to start, run zephyr application, then tear it down and start another application
- Low level messaging
    - Zephyr app is getting command on what to do over rpmsg
        - Demo both user space and kernel space rpmsg

- Higher level services
    - Zephyr app is using virtio file and socket connections to log data, dump memory, get time, etc.
- Throughput test
    - One command is to dump a big memory area back to Linux with time measurements
    - Perhaps have different ways of doing this:
        - rpmsg with zero copy (just send the pointer)
        - virtio file write
        - rpmsg with copying
- Latency test
    - Similar to throughput test, but using a small amount of data and instead measure round-trip time
- Easy to expand to show case other features by adding new “commands” from Linux app
- Other things to add in the future:
    - At some point we can add Xen as a hypervisor and showcase similar things as above, but between Xen guests as well as from Xen guest to Cortex R/M
    - Add a Trusted Execution Environment (OPTEE) configured by S-DT and doing some interactions

Tasks:

- Porting Zephyr, OpenAMP-rp and App Services stack to the target boards
    - OpenAMP-rp: remoteproc and rpmsg
    - App Services stack: KVMTools, VirtIO, etc.
    - Potentially harmonize the VirtIO solutions in rpmsg and KVMTools
- Use upstream QEMU as a target
- Lopper to produce device trees that Zephyr can consume
- Incrementally write the demo applications
- Integrate the different parts
- Document the flow
- …

Resources:
TBD, including potentially use contractors

Project management:
Potentially this can be a LITE project