How to make it easier for tools to deal with new languages, runtimes, and libraries?

We need software-defined events that originate from libraries, runtimes, or application layers.

General Ideas/needs:

- Define set of events/abstractions tools want to know about
- Define set of callbacks for these events
- Provide an abstraction layer through PAPI
- Avoid fixing/enforcing an API, but enable event flexibility for the low level layers and event querying for the higher layers

Events should not be just counters, more info is desired. Code centric information is somewhat well understood, but data centric views are less well understood. Examples for info desired (for an event):

- Code location
- Data array (src/dest, name and indexes)
- Thread id
- Stack trace
- Arbitrary data (let the library developer choose)

Example mentioned:

OCR: Need context of data operating on and thread it is operating

Need to support stacking/nesting of abstractions

Nesting can be supported/detected through the meta-data that is attached to events

Push/pull: Callback vs. query interface

Callbacks might not be desirable for library developers because they can introduce too much execution overhead, but query mechanisms add more burden in development and memory overhead.

Event frequency makes a lot of difference. An event that happens a thousand times a second, you don't want to call callbacks for it, you want to allow the higher layers to query it.

Static data:

How to handle static data (i.e. generated by compiler during compilation)?

- Provide runtime query
- Provide pre execution/post execution
- Define a file format for storing such data statically

Example related to collection of static information: Static markers (from red hat) sdt.h

https://sourceware.org/systemtap/wiki/UserSpaceProbeImplementation
https://sourceware.org/systemtap/wiki/AddingUserSpaceProbingToApps

And for example, using them from gdb:
https://sourceware.org/gdb//onlinedocs/gdb/Static-Probe-Points.html

Path forward: Possible Approach for getting things started:

Find a good example from a given tool, language, abstraction

ompt and mpit were discussed as examples of success stories