

# Functionality Slices Working Group

October 17, 2019

# Why?

- ▶ **Problem:** Given the size and structure of the PMIx standard document, it can be difficult to find the PMIx components necessary for a given use-case
- ▶ **Goal:** Provide a mechanism for focusing in on the aspects of the standard that are of interest to a particular user/use-case
- ▶ **Proposed solution:** compile a list of high-level use-cases and the PMIx capabilities required to support them

# Areas for Improvement

- ▶ It's not a quality problem, but quantity one
  - ▶ ~140 interfaces across ~340 pages
- ▶ The document is written primarily as an API reference
- ▶ Some interfaces "wear lots of hats"
  - ▶ `PMIx_Get`, `PMIx_Job_Control`, etc

# Audience & Questions We Want to Answer

## Users of Client-Side Interfaces

- ▶ I currently do X with some other interface/middleware, how do I do it with PMIx?
- ▶ I currently use the baseline PMIx functionality to do X, are there other (better) ways to do it in PMIx?
- ▶ I have some completely new use-case, can I mix-and-match existing PMIx functionality to meet my needs?

## Users of Server-Side Interfaces

- ▶ What server-side interfaces are necessary to support a given client interface or high-level use-case?
- ▶ What client-side interfaces are necessary to support a given high-level use-case?

# Goals

## Short-term:

- ▶ Use-Case Milestone 1 (Jan 2020):
  - ▶ bootstrap (business card exchange)
  - ▶ debuggers / tools
- ▶ Use-Case Milestone 2 (End of Q1 2020):
  - ▶ instant-on
  - ▶ hybrid programming models
  - ▶ workflow management
  - ▶ application management

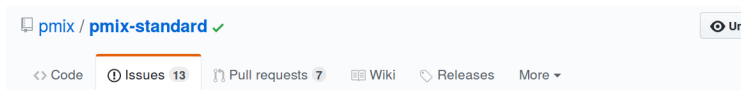
# Goals

## Long-term:

- ▶ Identify and document emerging use-cases:
  - ▶ tiered storage
  - ▶ power control
  - ▶ groups/flexible allocations
  - ▶ fault tolerance
  - ▶ cross-version compatibility
- ▶ Determine how to handle “miscellaneous” interfaces
- ▶ Document which resource managers support which use-cases
- ▶ Add interfaces into the standard for querying "use-case support"

# Work Completed So Far

- ▶ Went through each RFC, documenting the interfaces and use-cases covered
- ▶ Aggregated the RFC notes into rough slices
- ▶ Analysis of what interfaces were not covered by RFCs
- ▶ Drafts of Debugging & Bootstrap Use-Cases
  - ▶ Currently "issues" in the pmix/pmix-standard GitHub repo



## Debugging #216

Open spophale opened this issue on Sep 4 · 2 comments



spophale on Sep 4 · edited ▼



### Brief Description

This use case is an attempt to distill out the features/extensions requested in the RFCs that are related to debugging. We have identified parts of PR23 (Co-located process launch for debuggers), RFC0010 (MPIR-like query), RFC0002 (event pub/sub), RFC0022 (Environmental Parameter Directives for Applications and Launchers), and RFC0025 (IO Forwarding) under this category.

### Use Case Details

# Call for Participation

## Works In Progress

- ▶ General structure and template for Use-Cases
- ▶ Hybrid Programming Models & Instant-On Use-Cases
- ▶ Coverage Analysis

## Help Wanted

- ▶ Identifying use-cases that we've missed
- ▶ Reviewing documented use-cases
- ▶ Documenting identified use-cases



# Call for Participation

## How to Join

- ▶ Subscribe to our Google Group mailing list
  - ▶ pmix-forum-wg-func-slices
- ▶ Weekly call
  - ▶ Wednesdays @ 9am PT / 12pm ET
  - ▶ Webex info sent via mailing list
  - ▶ Email me for a calendar invite

## Rewards

- ▶ Eternal love, gratitude and admiration from the PMIx community
- ▶ If you ever close by, I'll buy you a beverage of your choosing

# Discussion

- ▶ Would you want to introspectively query the runtime for use-case support?
- ▶ Would you want a compatibility table for Use-Cases x Resource Managers?
  - ▶ One way that the community represented by the ASC could track PMIx compliance
- ▶ What is the best way to publish this work?
  - ▶ Separate repository of documents like RFCs
    - ▶ Separate directory within pmix-standard repo
  - ▶ Re-order standard document into functionality slices
    - ▶ Overlap goes into a “core” slice
  - ▶ Add an appendix to the standard with the interface groupings that links back to the full interface definitions
  - ▶ Dynamic website that lets you switch between interface-centric vs use-case-centric groupings