



Rocprofiler-SDK

Benjamin Welton
on behalf of AMD ROCProfiler Team

Scalable Tools Workshop
August 12, 2024

AMD 
together we advance_

What is Rocprofiler-SDK?

New GPU Performance Tool Library

- Provides APIs for:
 - Tracing (HSA, HIP)
 - GPU HW PMC (Performance Counters)
 - Collected on either a per-dispatch or device wide basis (agent profiling)
 - PC Sampling (HW supported on MI300, Trap based MI2XX)
- Replaces rocprofilerv1, rocprofilerv2, and roctracer

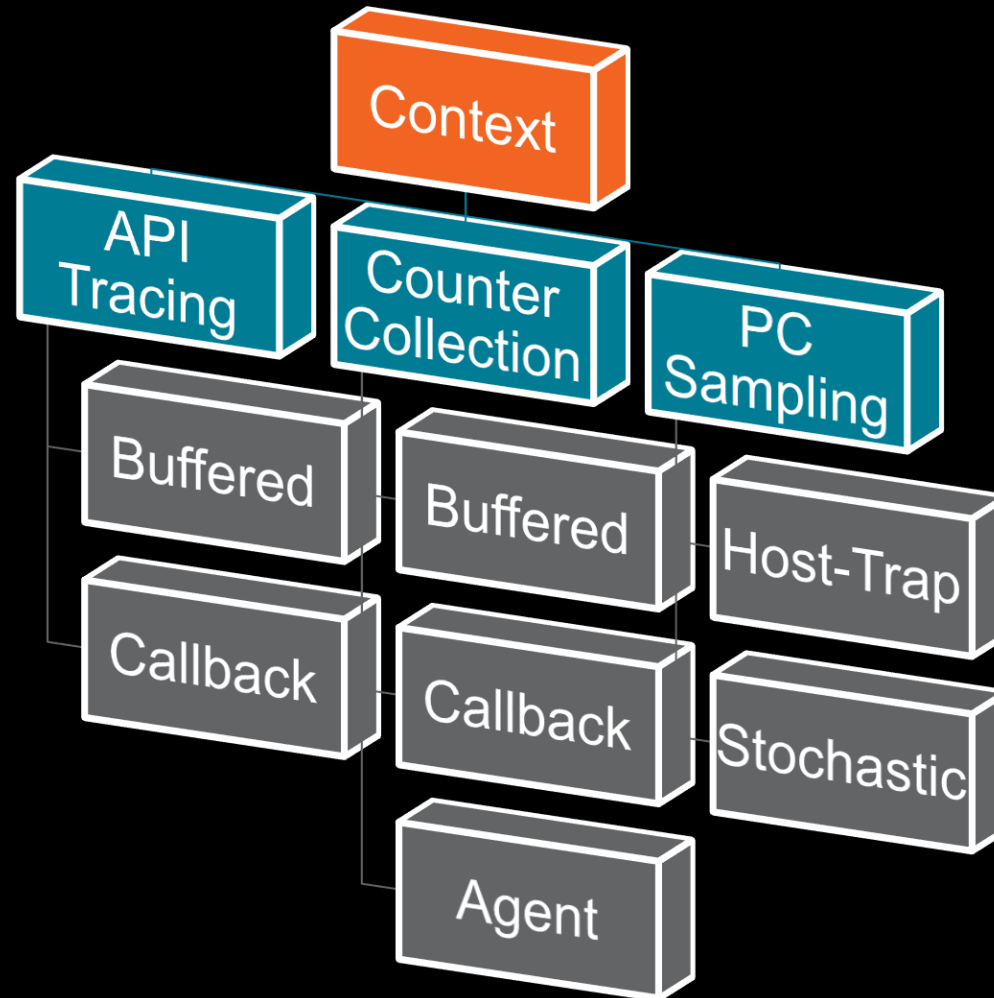
Rocporfiler-SDK Design Goals

- Simplify Design and Usage
 - Removed the need to use multiple different SDKs to collect common performance data
 - Revamped the interface to make it easier to setup
- Lower Profiling/Tracing Overhead
- Improve Stability/Accuracy
 - Improve testing to catch issues
 - Add tests to verify values obtained from tracing
- Expand Feature Set
 - PC Sampling, better device HW PMC counter support, etc.

Community Feedback Lead Development

- Wanted to include the tool community in development of rocprofiler-sdk
 - You are our customers, and we wanted to address your problems
- Avoid black box development where things change without explanation/warning
- Active role in helping port over community written tools
 - Personal hack-a-thons with several tool development teams to migrate their code and get feedback
 - HPCToolkit
 - Caliper
 - TAU
 - PAPI
 - Score-P

Core Rocprofiler-SDK Services



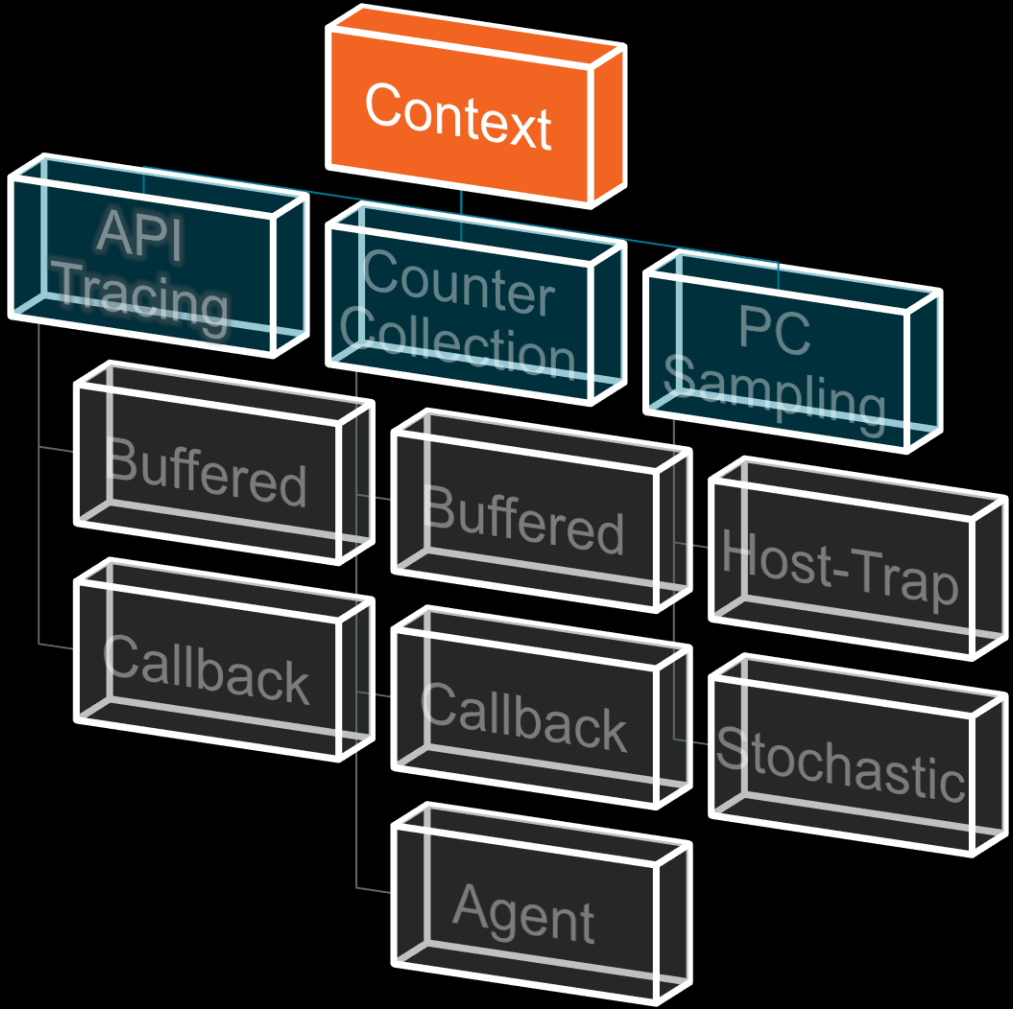
Core Rocprofiler-SDK Services

Context Immutable object where the tool defines what services to use

Multiple services can be defined in a context (i.e. tool may want to use API Tracing with Counter Collection)

Multiple contexts can be created and turned on/off as needed

Allows tools to be notified early if incompatible options are present

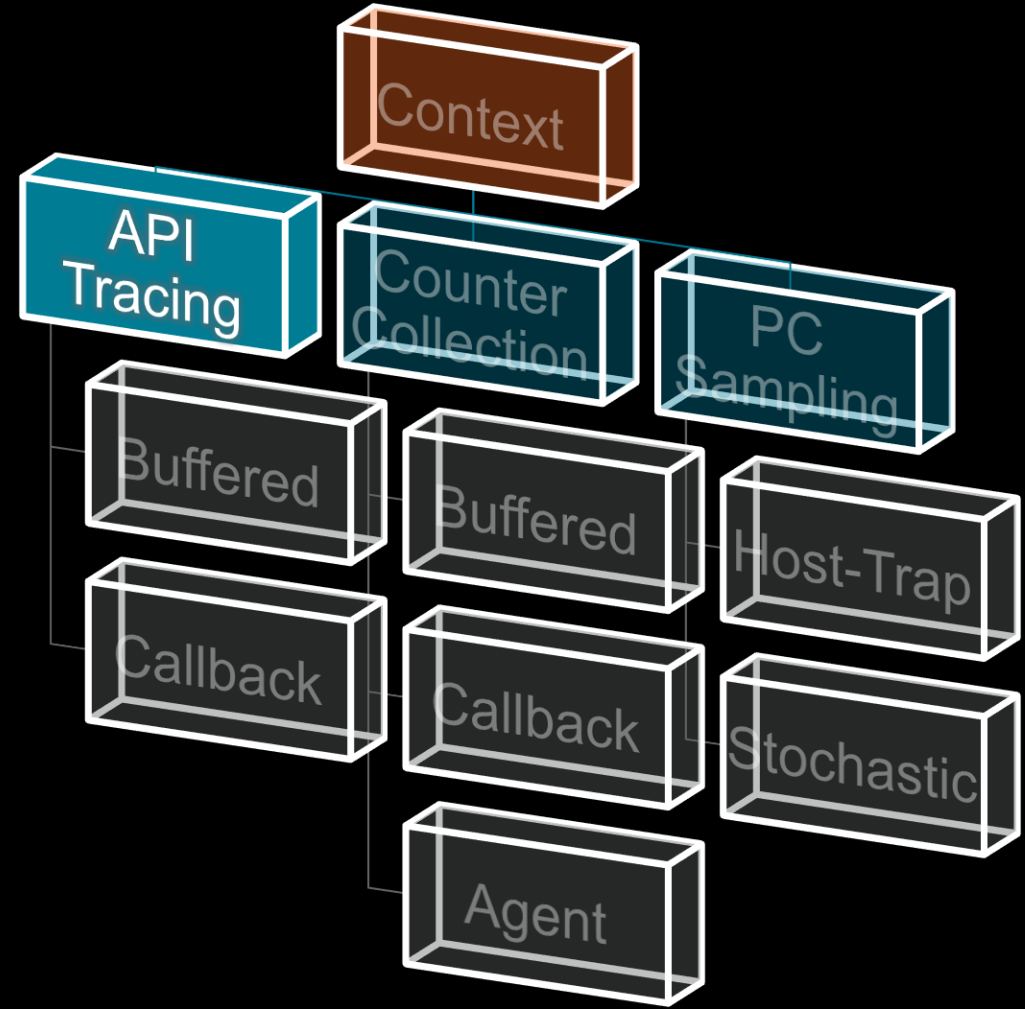


Core Rocprofiler-SDK Services

API Tracing Provides HSA and HIP callback tracing

Can ask for traces based on higher-level operations (such as trace calls that perform memory copies)

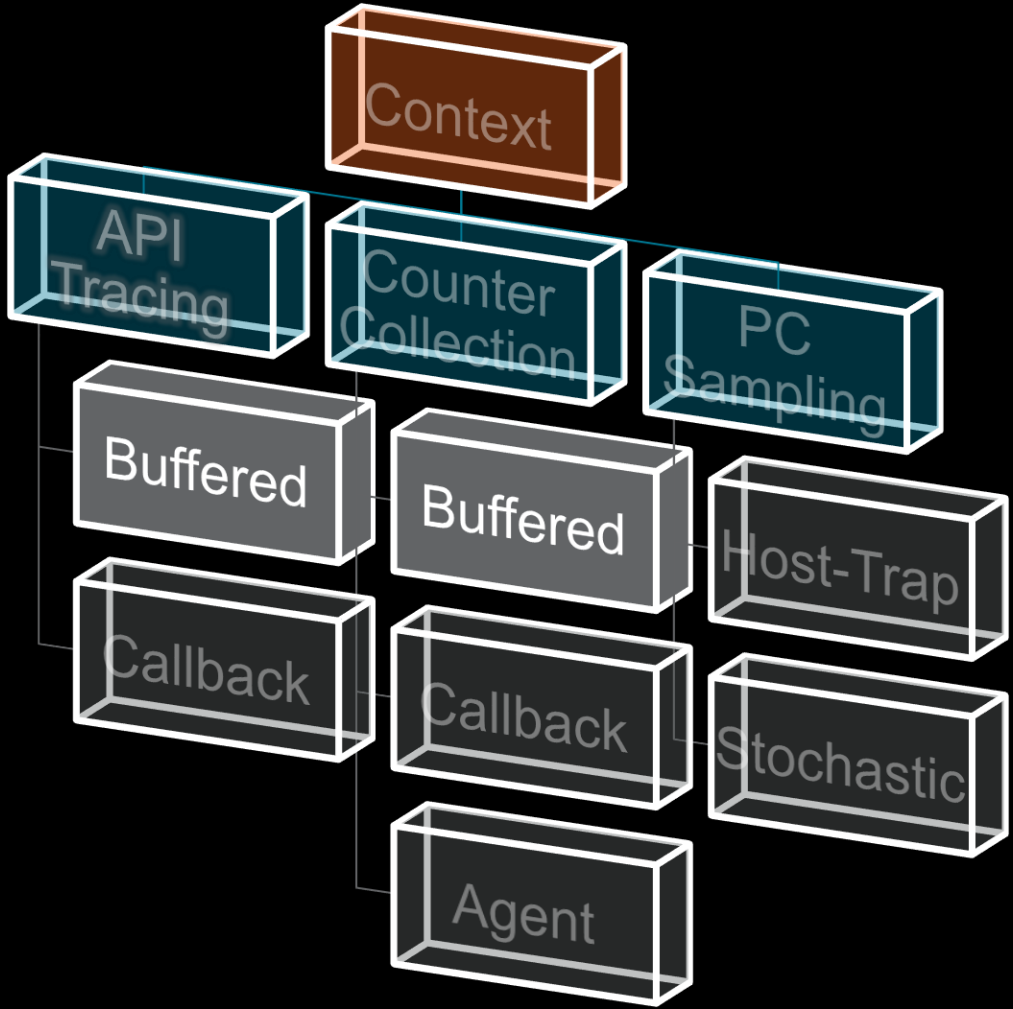
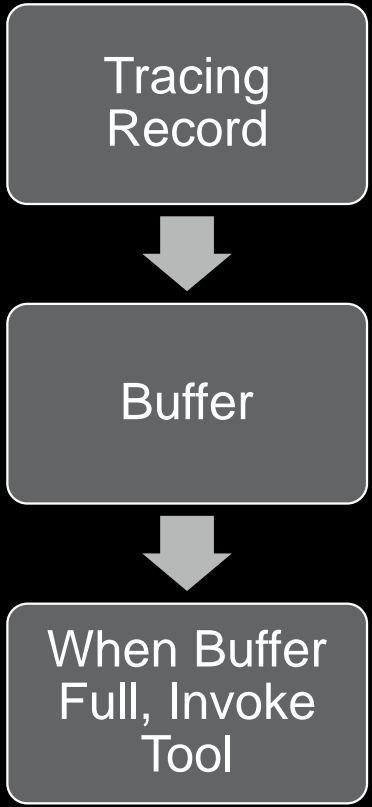
Multiple tools and contexts can perform tracing at the same time



Core Rocprofiler-SDK Services

Buffered Tracing

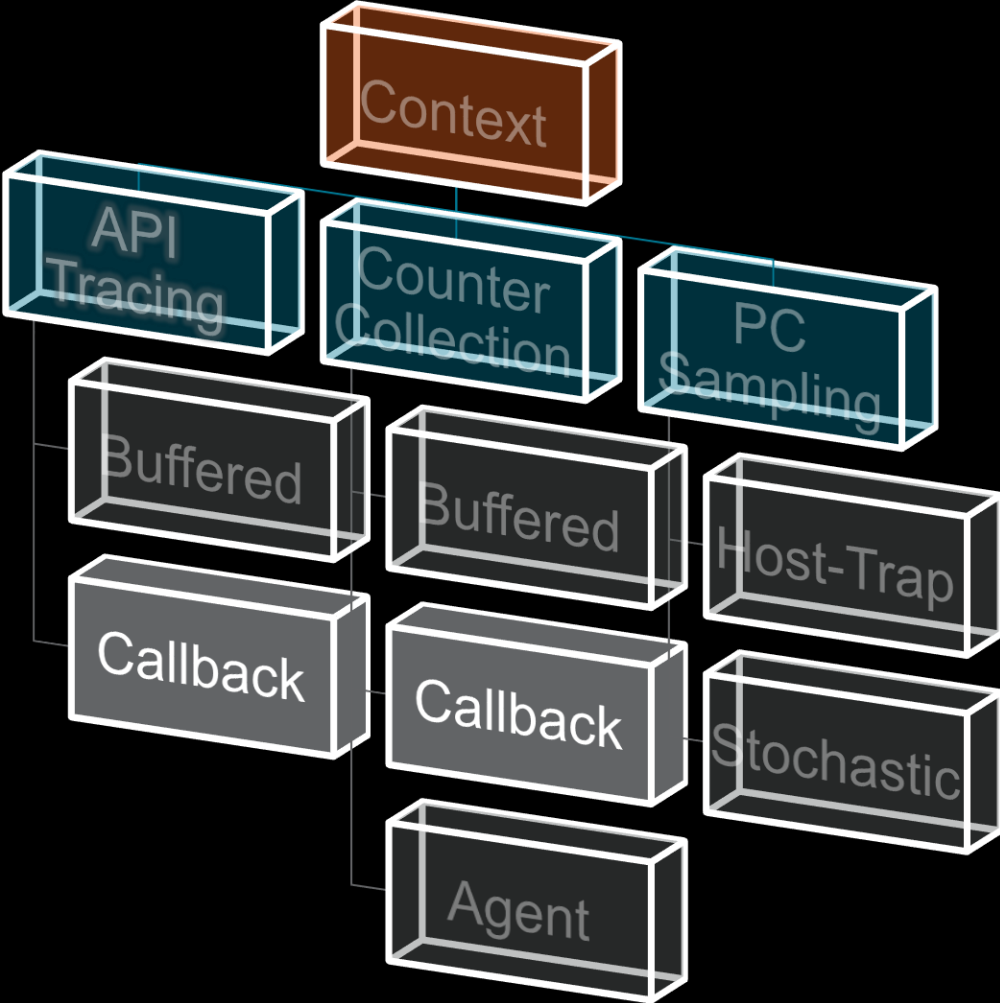
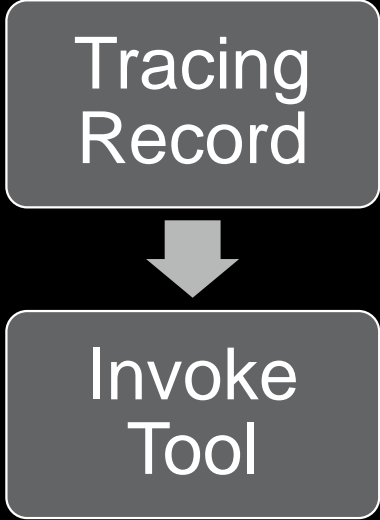
Stores records in a buffer, notifies tool when buffer is full



Core Rocprofiler-SDK Services

Callback Tracing

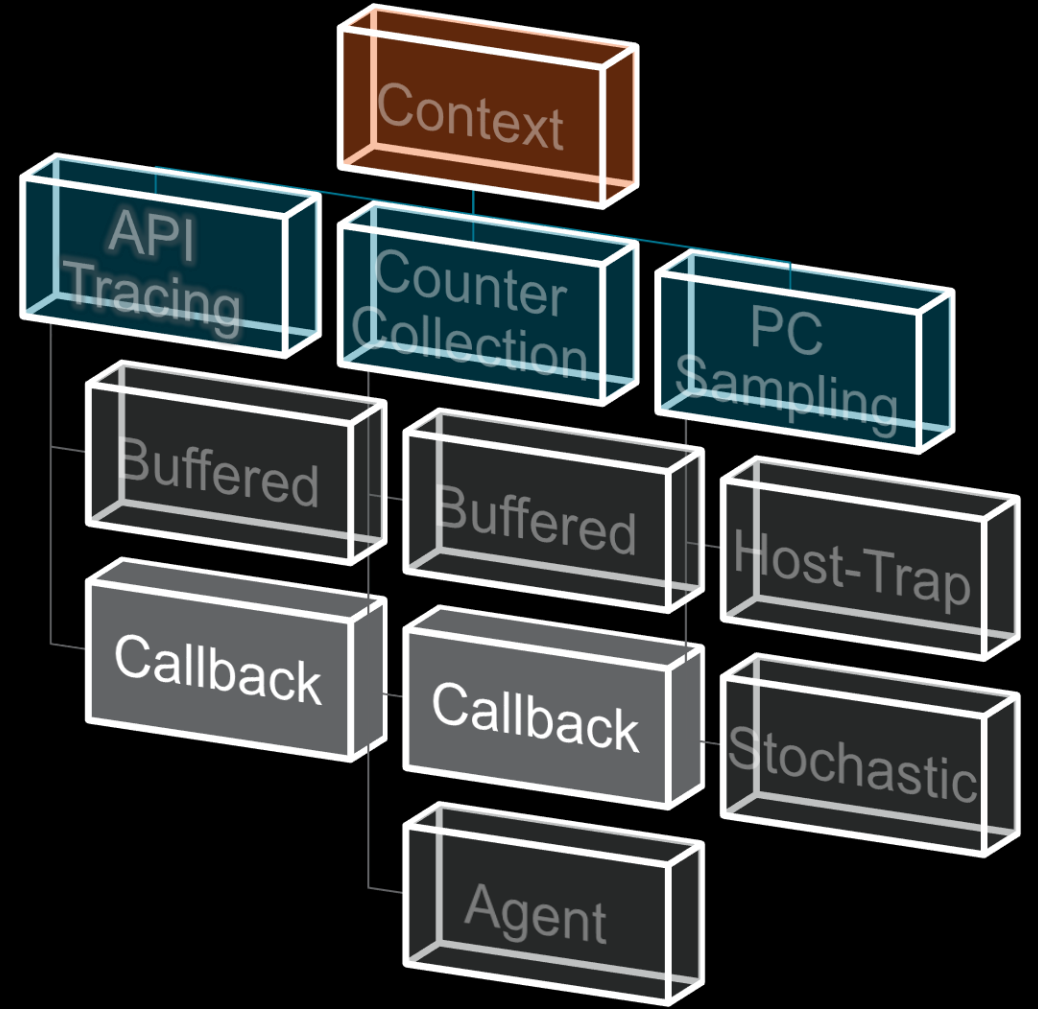
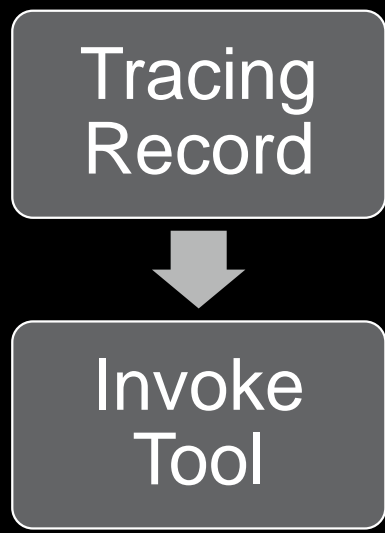
Immediately calls tool when function is invoked/completed



Core Rocprofiler-SDK Services

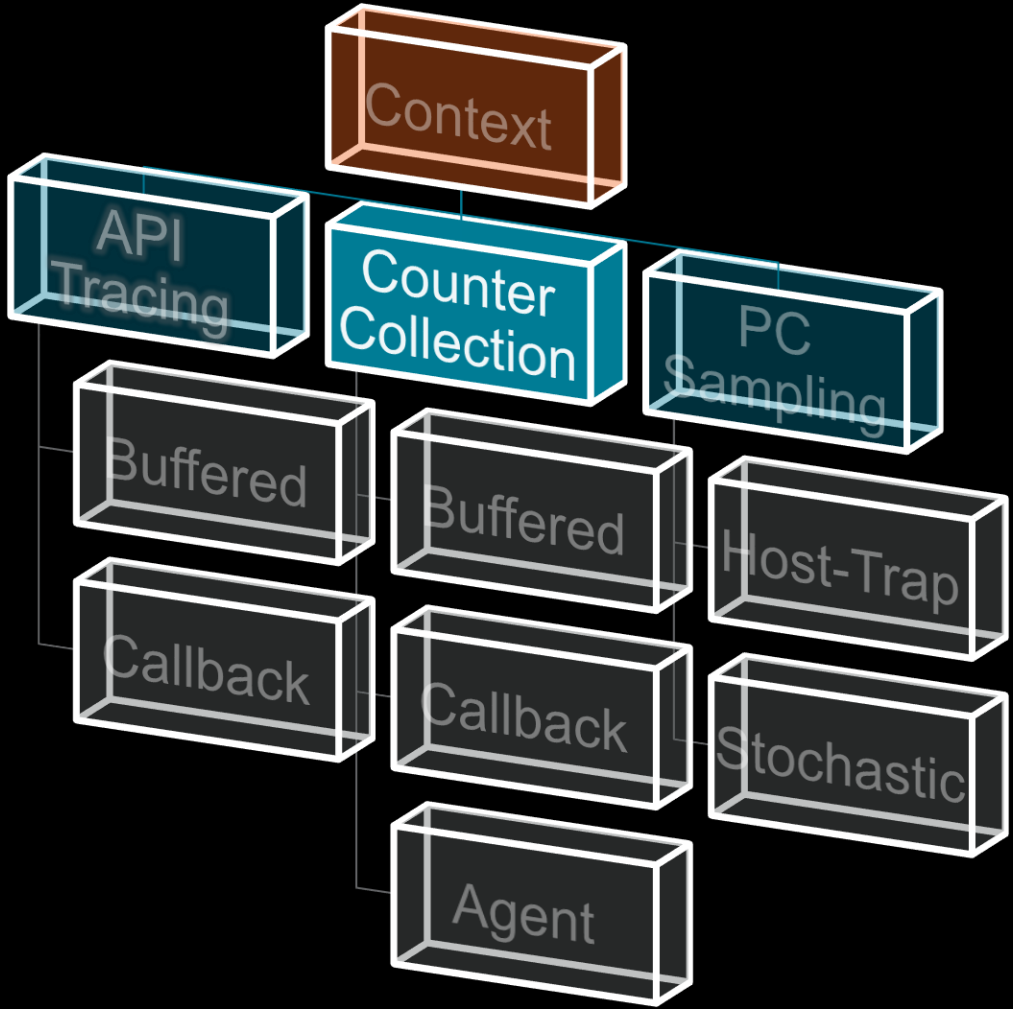
Callback Tracing

Immediately calls tool when function is invoked/completed



Core Rocprofiler-SDK Services

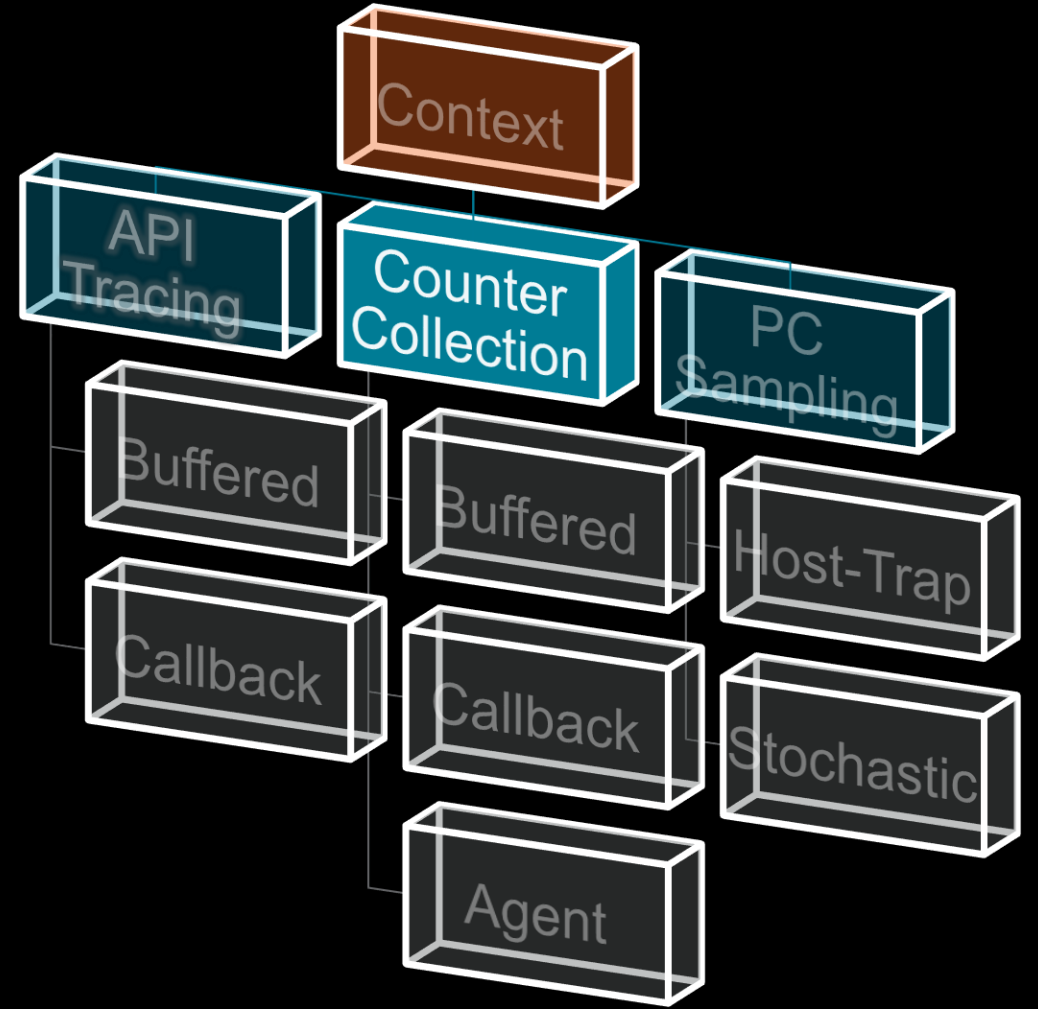
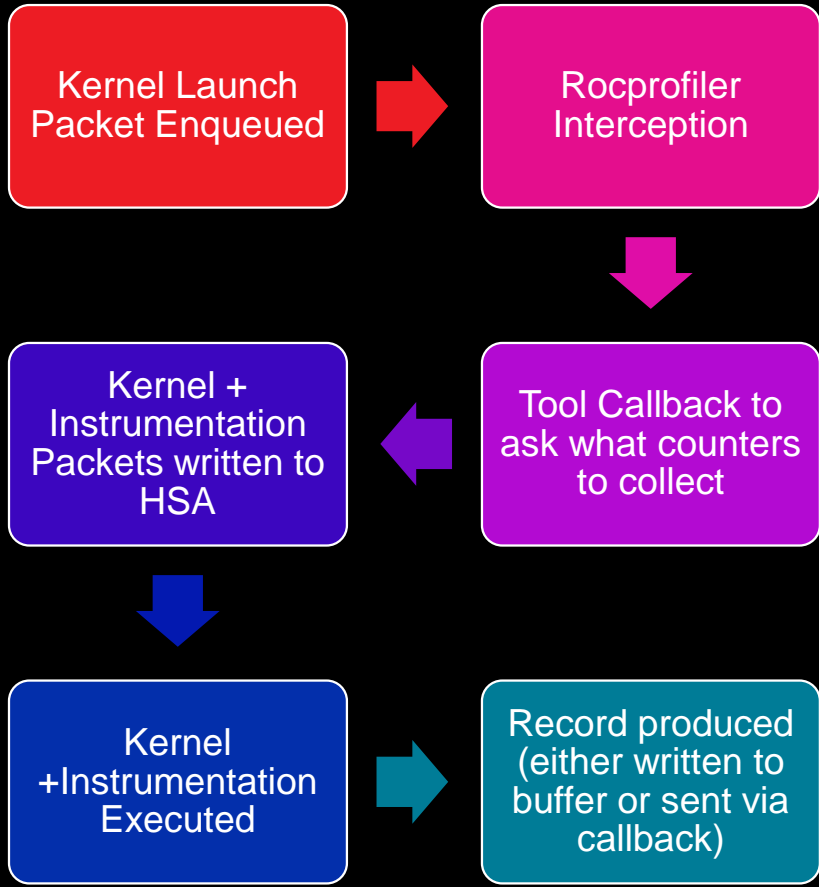
Counter Collection	HW PMC Counter Interface
Two modes:	
Dispatch based counter collection	
Agent based counter collection	



Core Rocprofiler-SDK Services

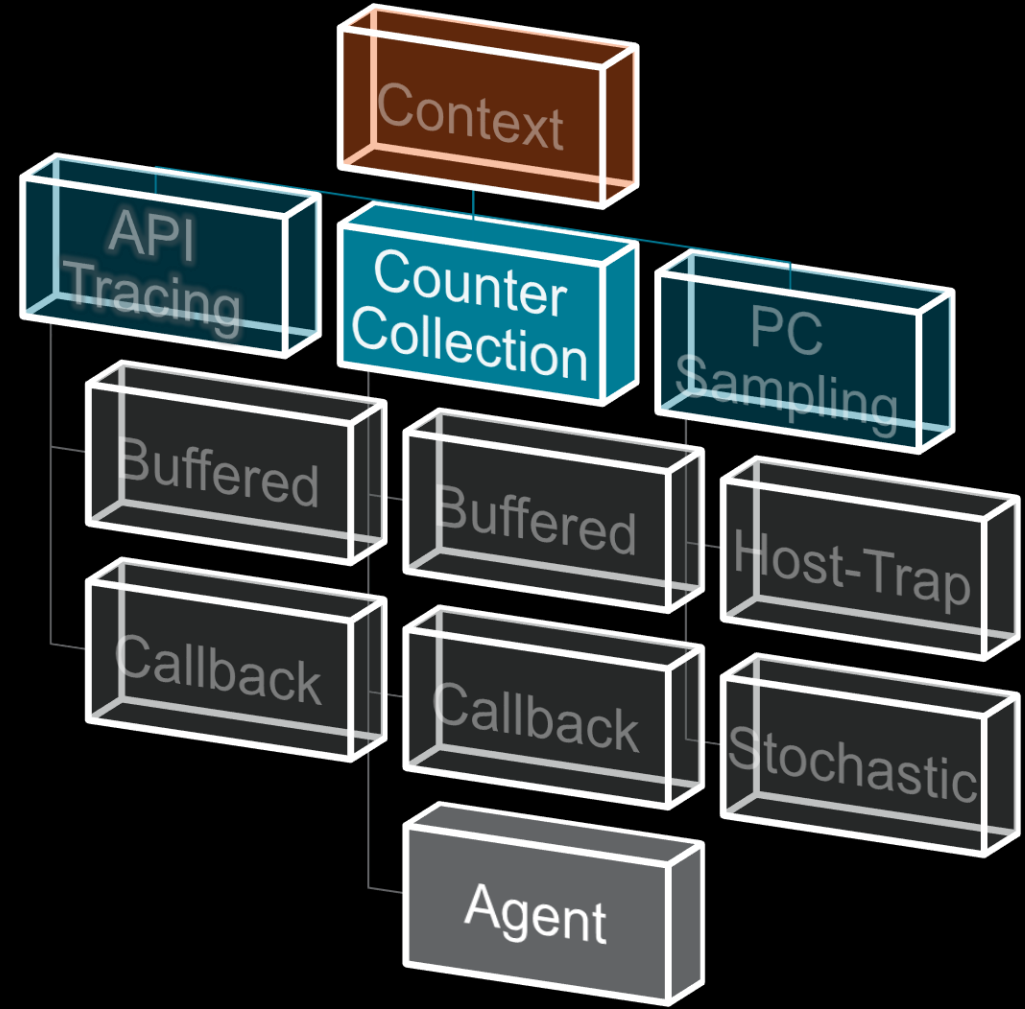
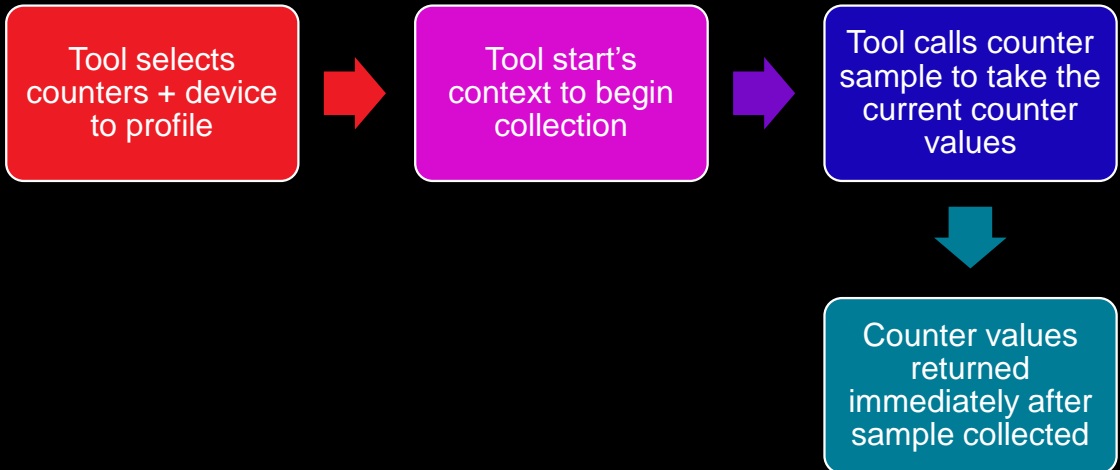
Counter Collection

Dispatch Based Counter Collection



Core Rocprofiler-SDK Services

Agent Based Counter Collection	Collection of counter values on the agent as a whole (not for specific kernel executions)
	Tool Responsible for manually starting/stopping/and sampling values (via API calls)
	Useful for System-Wide monitoring



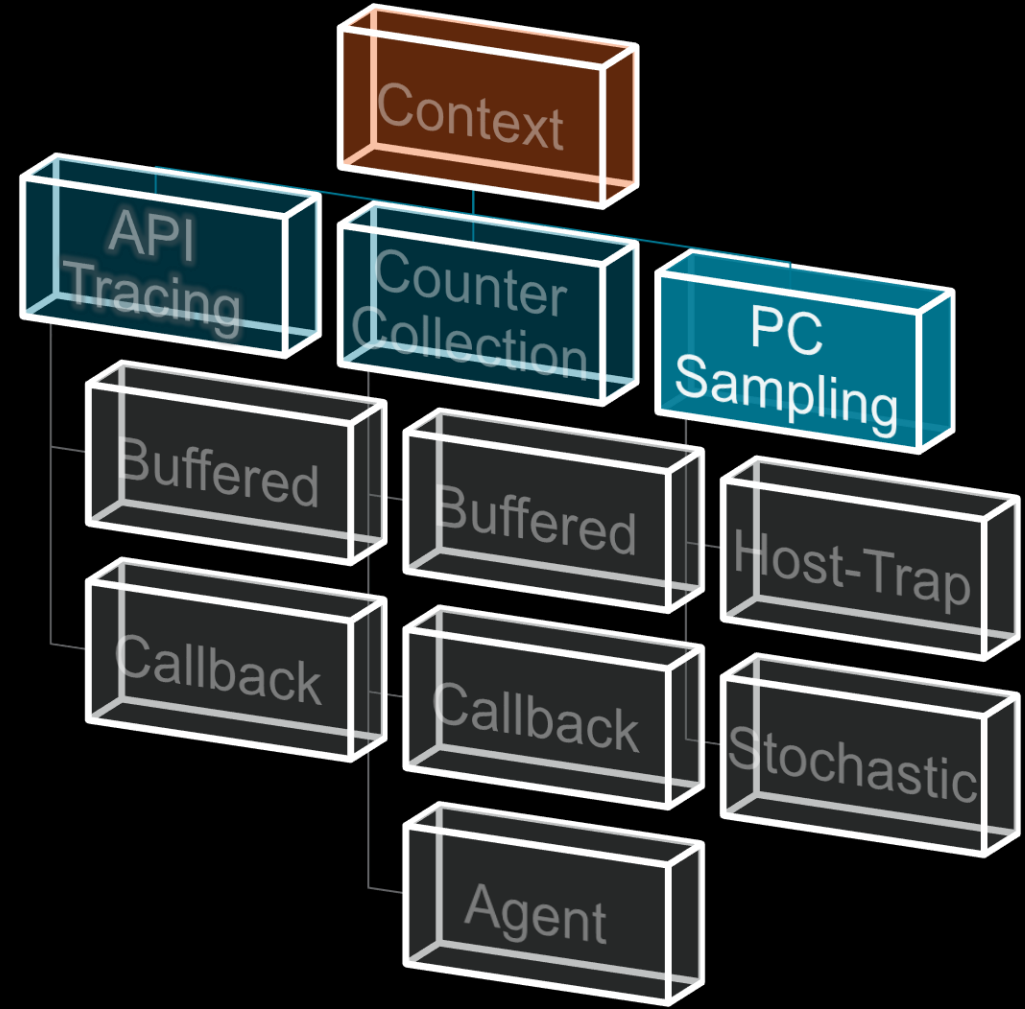
Core Rocprofiler-SDK Services

PC Sampling Allows collection of PC Samples of GPU Kernels

Two modes:
Host-Trap and Stochastic

Hardware Support is limited to MI2XX (Host-Trap Only) and MI3XX (Host-Trap + Stochastic)

Tool specified sampling and interval periods



Core Rocprofiler-SDK Services

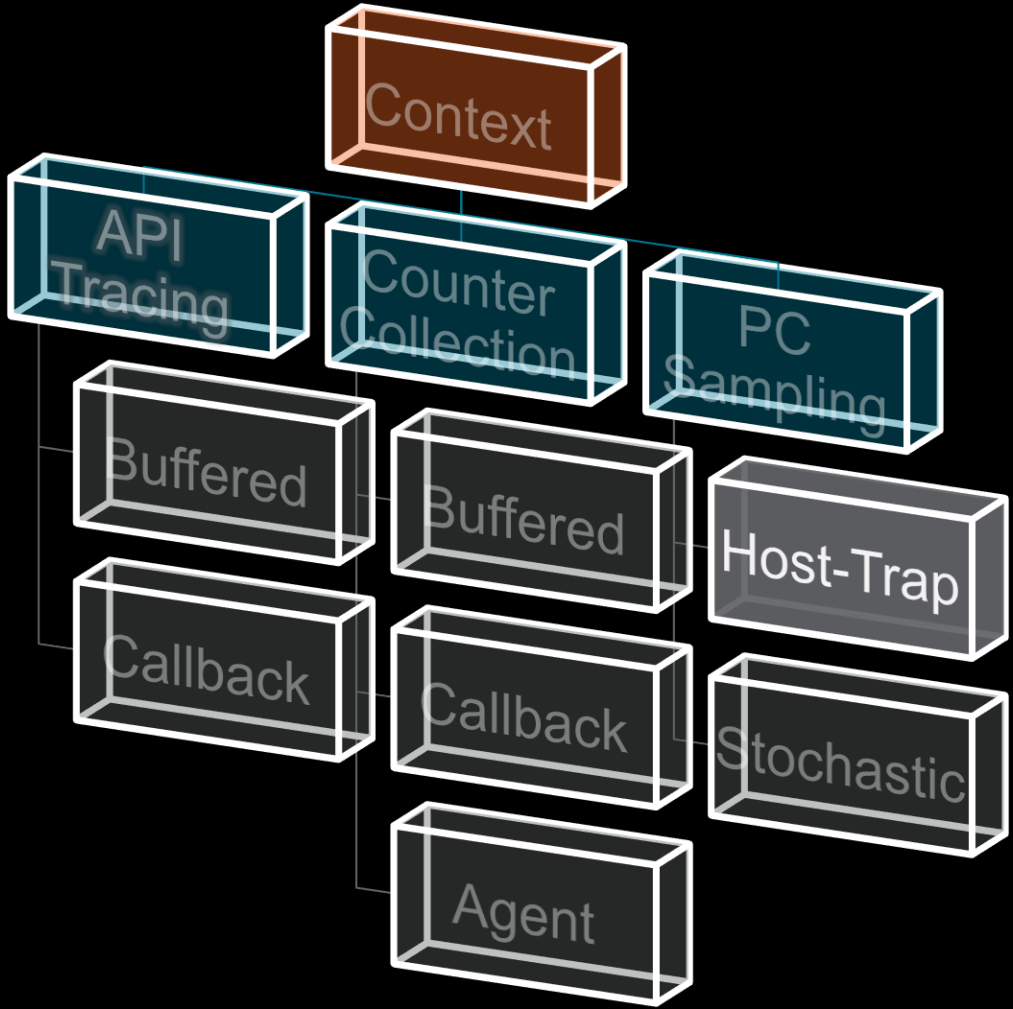
Host-Trap Sampling

Driver (KFD) creates a kernel thread to periodically inject s_trap into the GPU distributed sequencer (SQ)

Each wave is trapped in a round-robin manner, outputting PC addresses via PCI bus.

Data collected includes PC, exec_mask, timestamp, etc

Wave selection/trap issuance is done at Linux KFD level



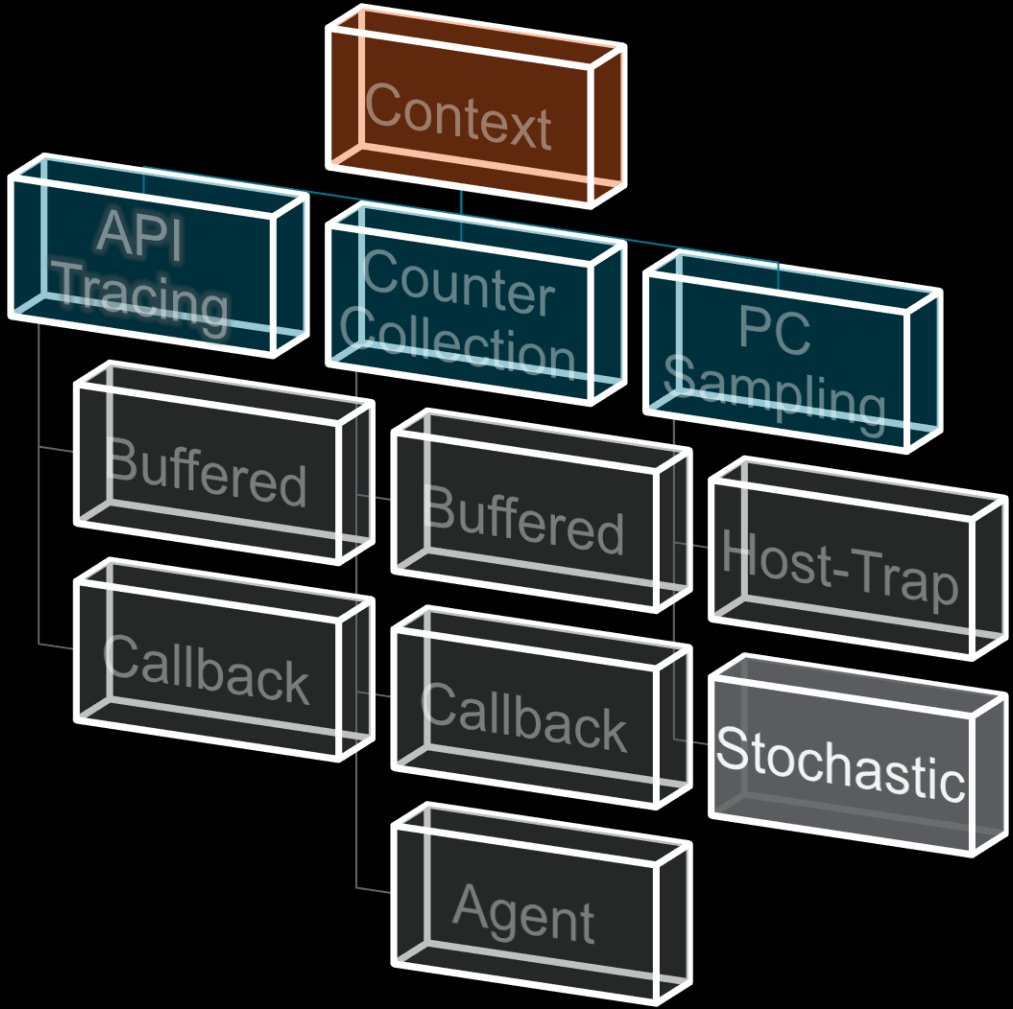
Core Rocprofiler-SDK Services

Stochastic Sampling




Similar to Host-Trap but instead uses a HW based solution to wave selection on GPU.

Lower overhead than host-trap based solutions (i.e. not linux kernel driven, but hardware driven).

Still a work in progress



Requested Features & Improvements From External Tool Developers

- General performance improvement 
- Dramatically improve reliability and stability between ROCm releases 
- Robust initialization for rocprofiler and tools
- Multi-tool Support
- Make external correlation IDs more useful
- Ability for tool to force initialization (instead of relying on rocprofiler to call tool)
- Notifications for when rocprofiler creates internal threads
- Thread pool for buffer callbacks
- Improved thread-safety, error handling, memory access correctness
- Eliminate memory leaks
- Improve counter collection performance 
- Topology/agent information without having to use (and initialize) HSA-runtime and/or HIP

Reliability and Stability Improvement Updates

- Code coverage
 - Tests: 87%
 - Samples: 75%
 - Total: 87%
- Integration testing framework validates output data produced by library / rocprofv3
- Strict static assertions ensuring runtimes (HSA, HIP, etc.) do not break tracing ABI
- **Shifting paradigm → decoupling profiling from runtimes**

Availability and Samples

- Rocprofiler-SDK beta is available now in ROCM 6.2.
 - Full release tentatively in ROCM 6.3 (caveat we reserve the right to change the API during the transition from v1/v2)
- Source code is open source:
 - [ROCm/rocprofiler-sdk \(github.com\)](https://github.com/ROCm/rocprofiler-sdk)
- Samples are included to show how to use rocprofiler-sdk in tools
- New version of rocprof (the tool) utilizing the SDK will also be available
 - New output mode like CTF and others will be supported

Copyright and disclaimer

- ▶ ©2024 Advanced Micro Devices, Inc. All rights reserved.
- ▶ AMD, the AMD Arrow logo, CDNA, EPYC, Instinct, Infinity Fabric, ROCm, Ryzen, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.
- ▶ The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.
- ▶ THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION

AMD 