DAMON:
Updates and Future Plans

SeongJae Park <sj@kernel.org>

Notices

- The views expressed herein are those of the speaker;
  they do not reflect the views of his employers
- The download link for these slides will be available at a reply to the original talk proposal (https://lore.kernel.org/damon/20230214003328.55285-1-sj@kernel.org/), which is also linked in LSFMM schedule google doc

https://twitter.com/sjpark0x00/status/1295387149018300419/photo/1

I, SeongJae Park <sj@kernel.org>

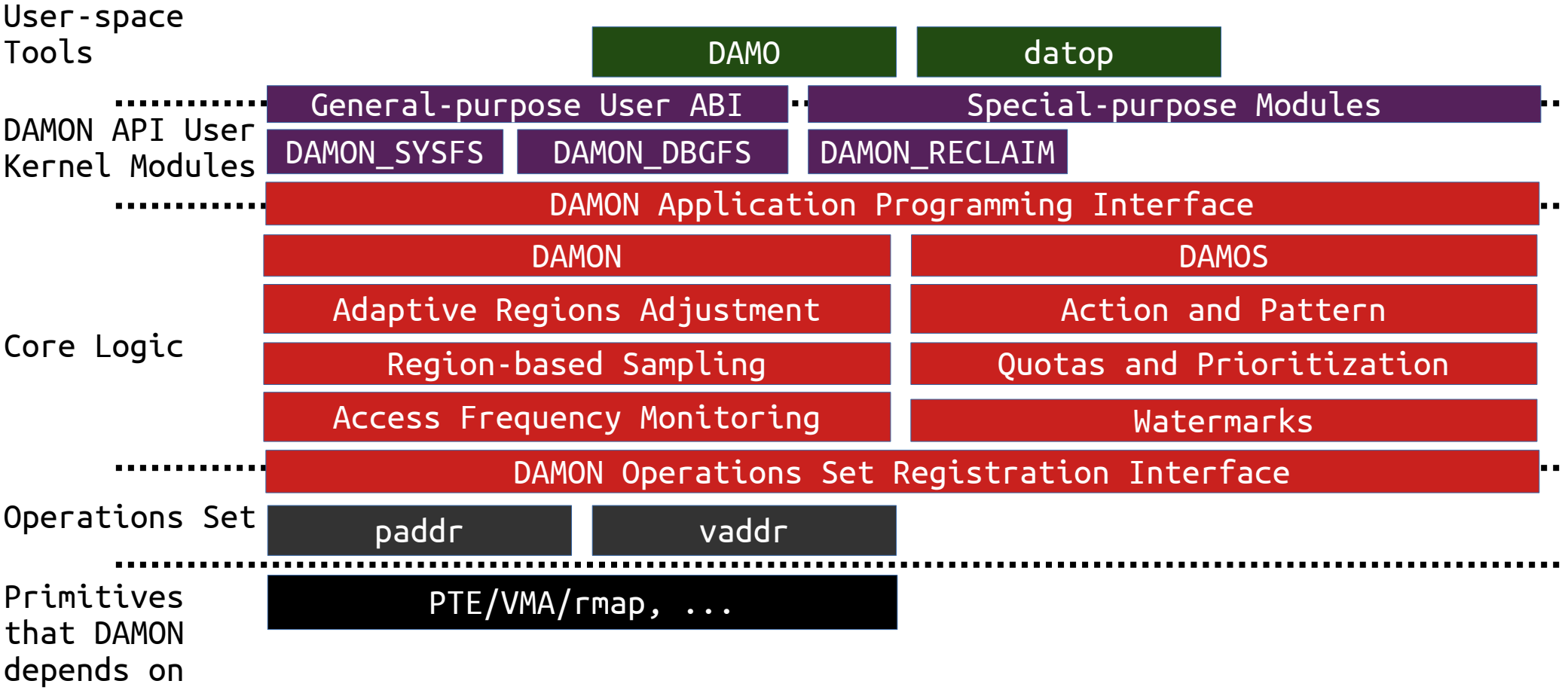
- Just call me SJ, or whatever better for you to pronounce
- Kernel Development Engineer at AWS
- Interested in the memory management and the parallel programming
- Maintaining DAMON

Overview

- Recap of DAMON development until LSFMMBPF 2022
- Updates of DAMON development until LSFMMBPF 2023
- Future Plans
  - Short Term (~2023)
  - Long Term (Not-yet-Scoped)
- Discussions

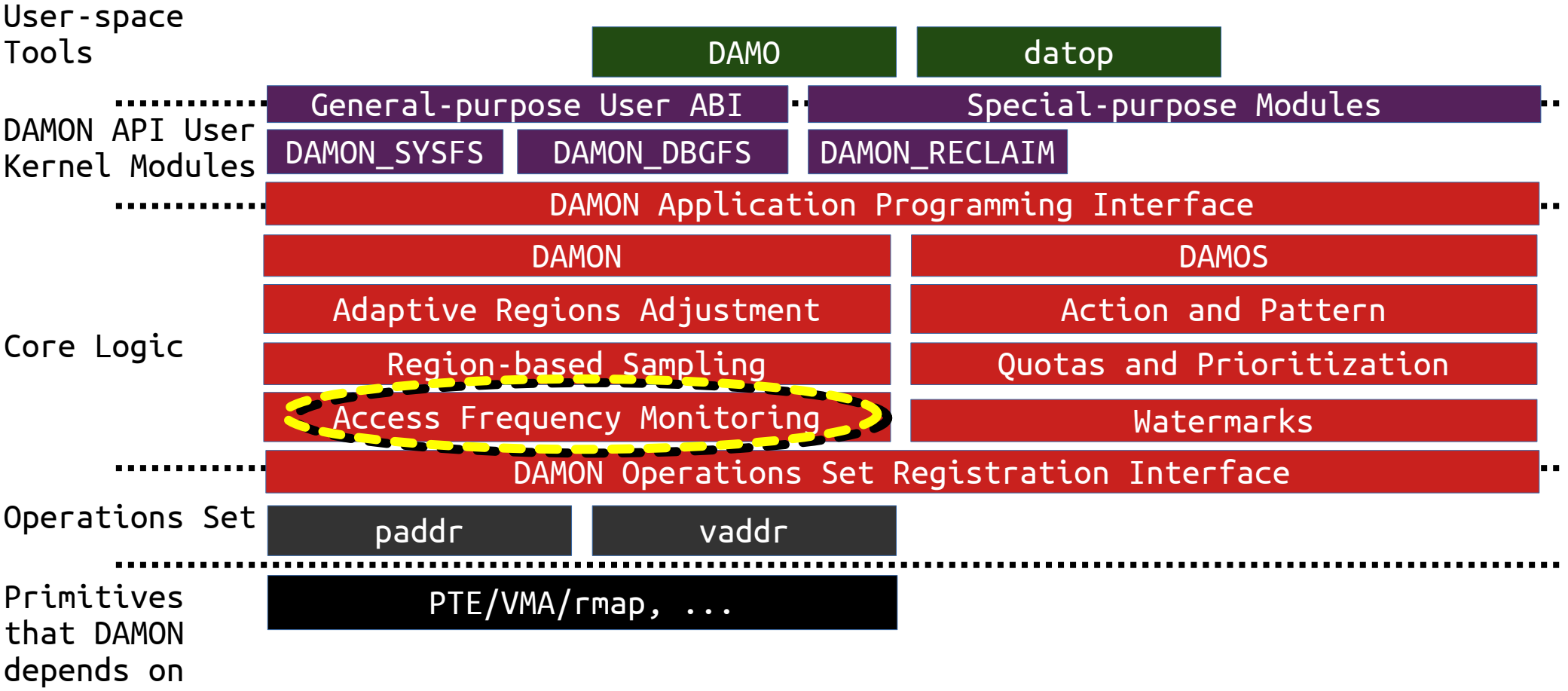
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Introducing again since this is the first DAMON talk at LSFMMBPF



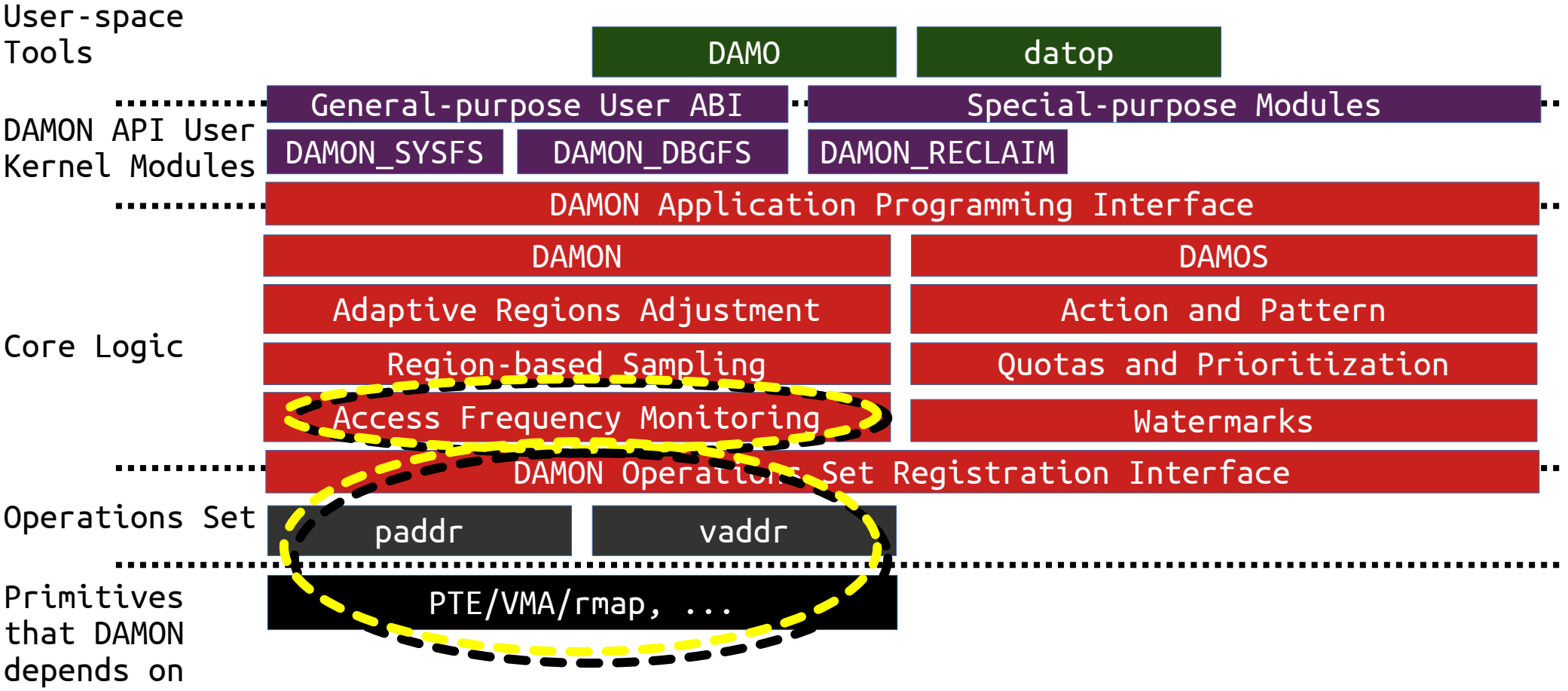
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



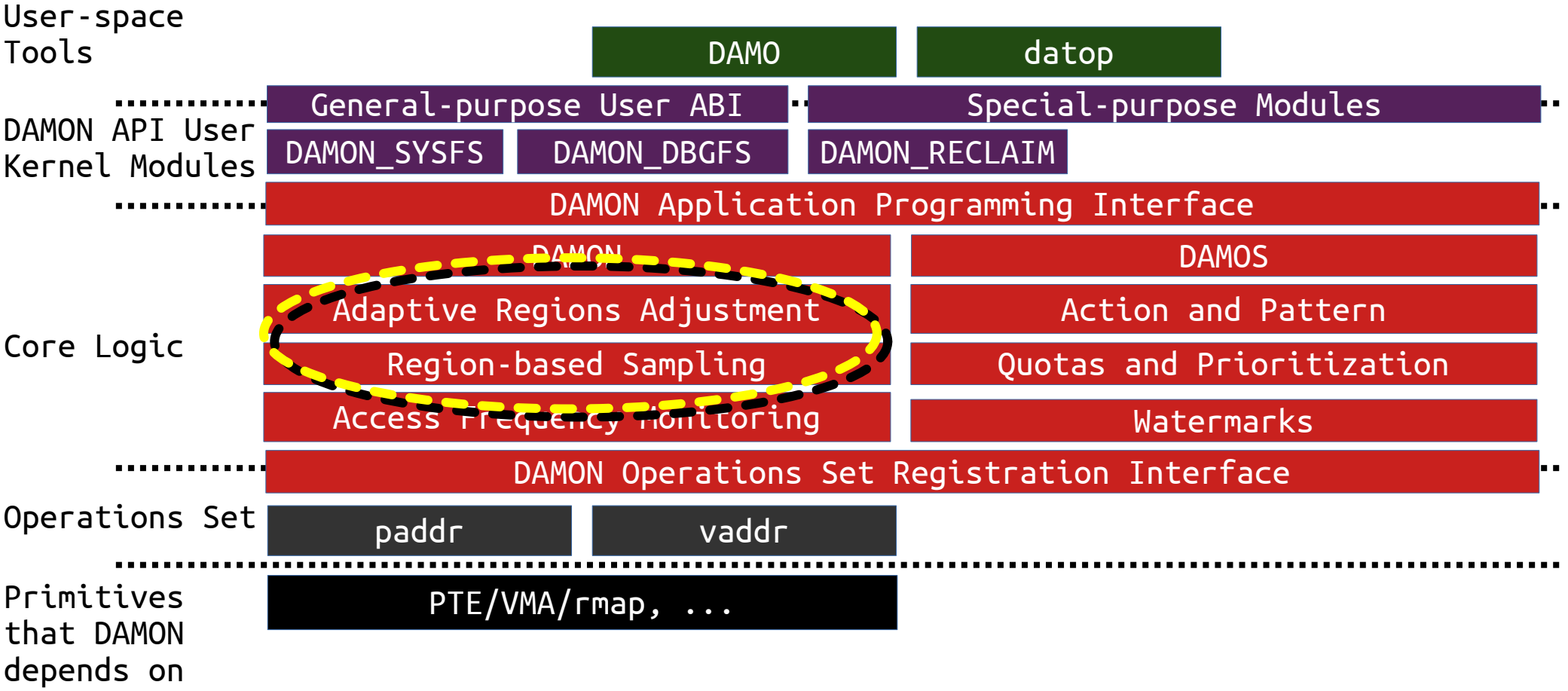
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



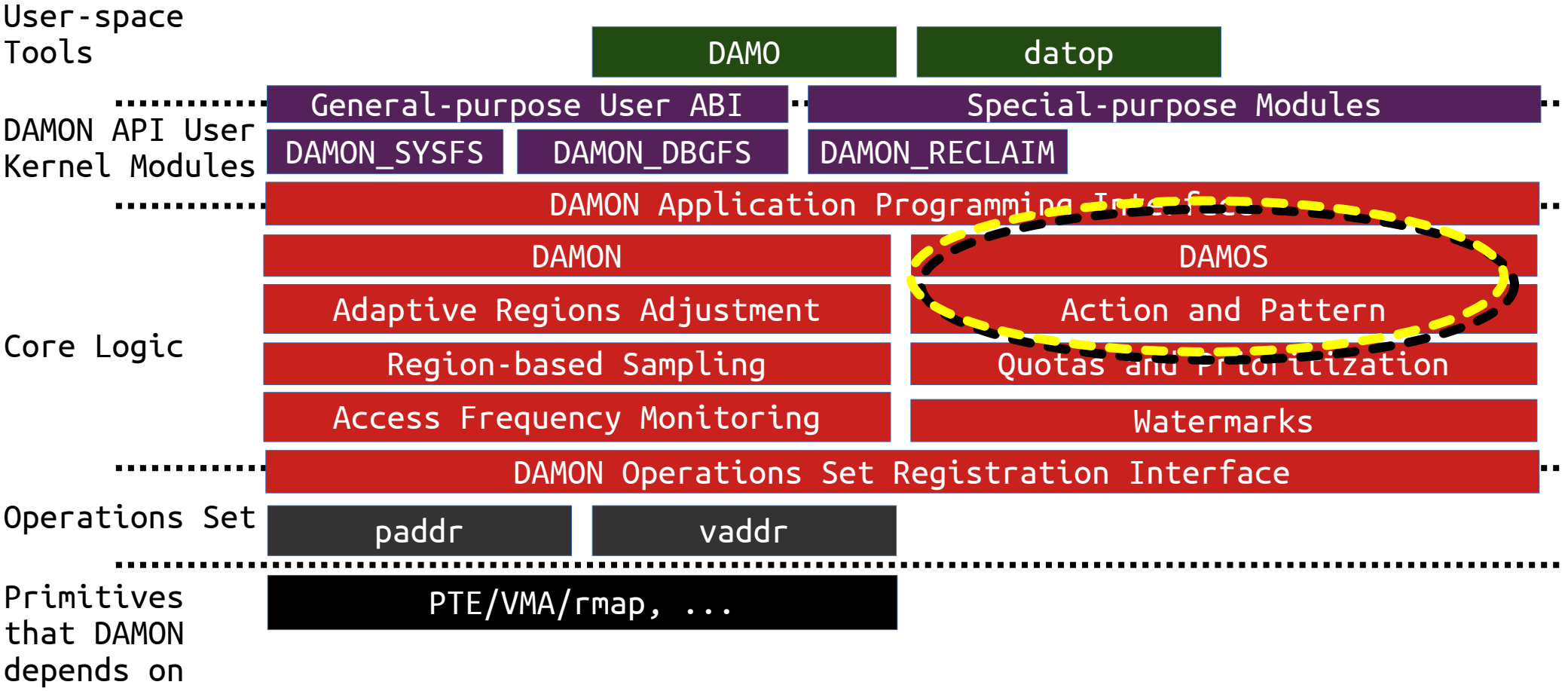
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



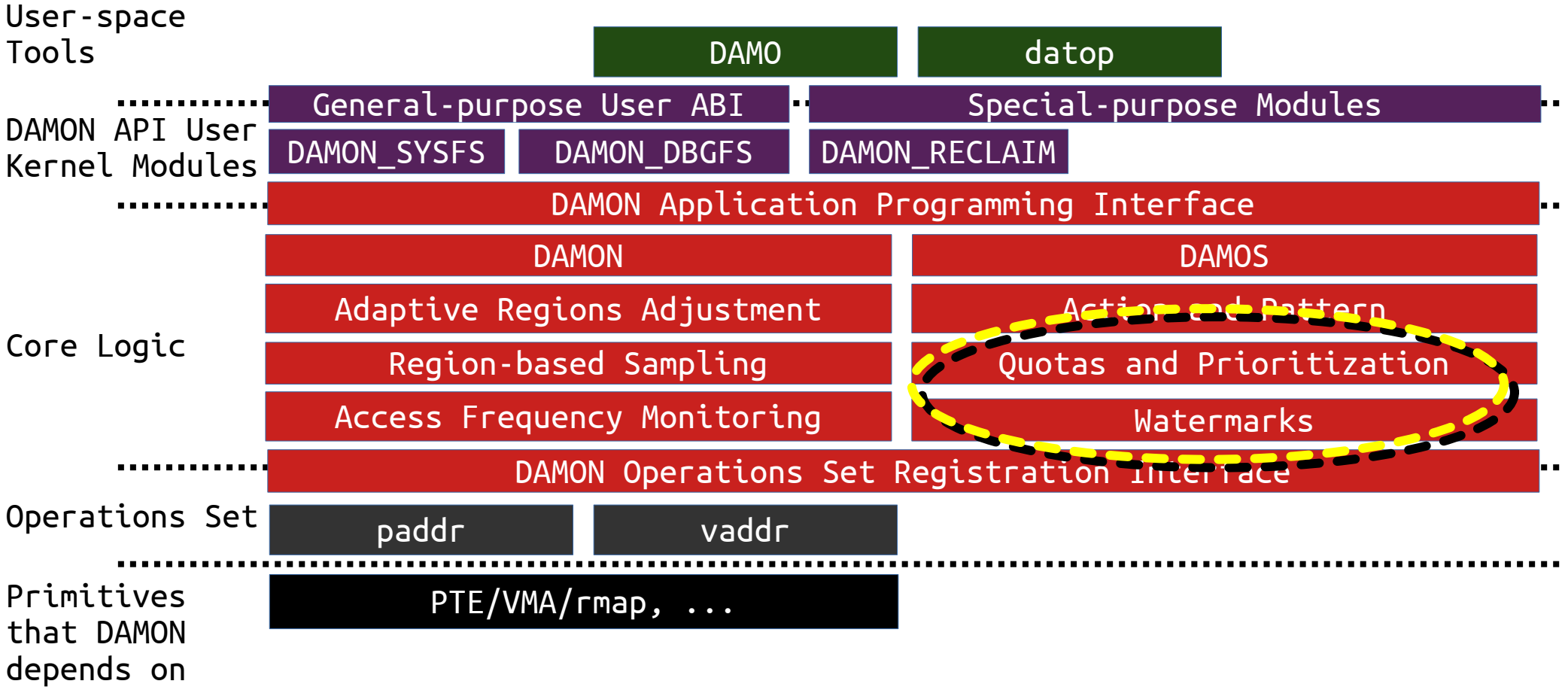
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



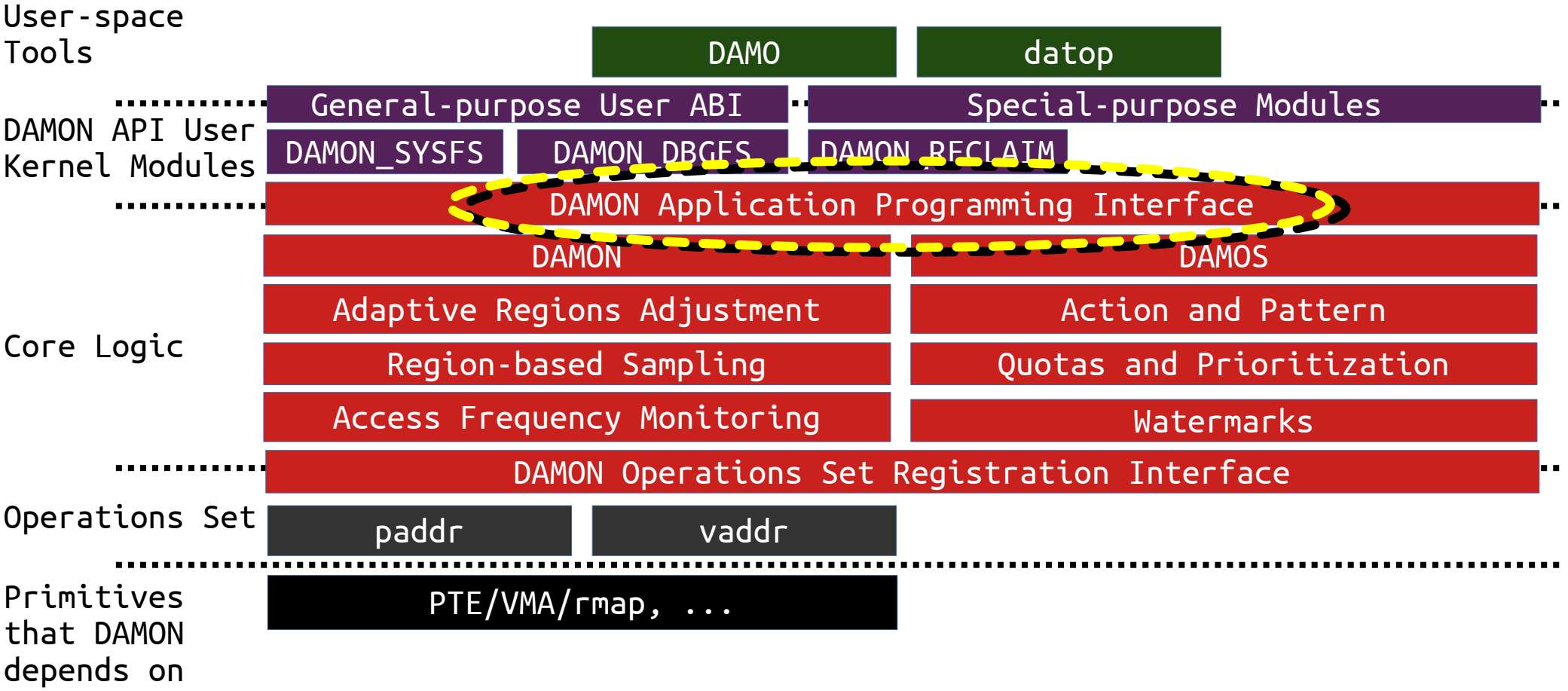
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



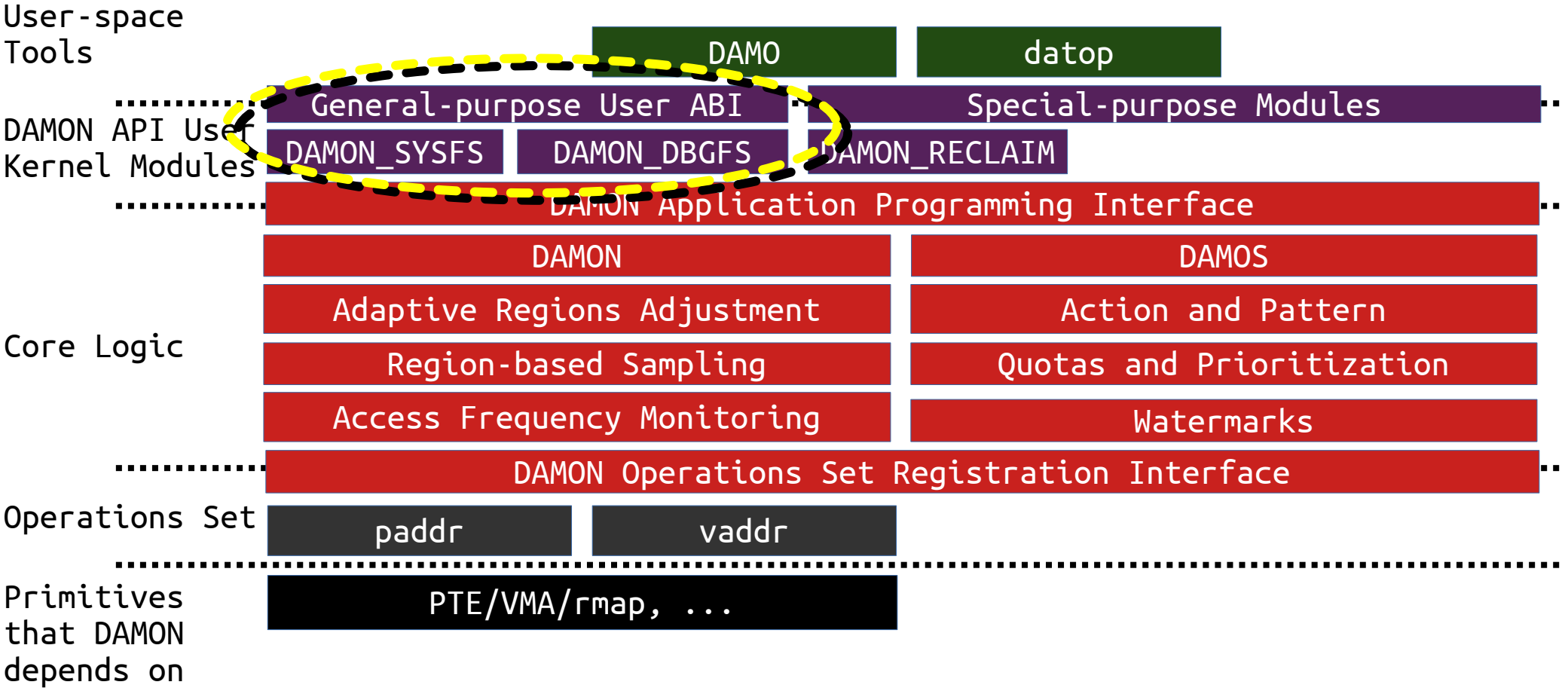
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



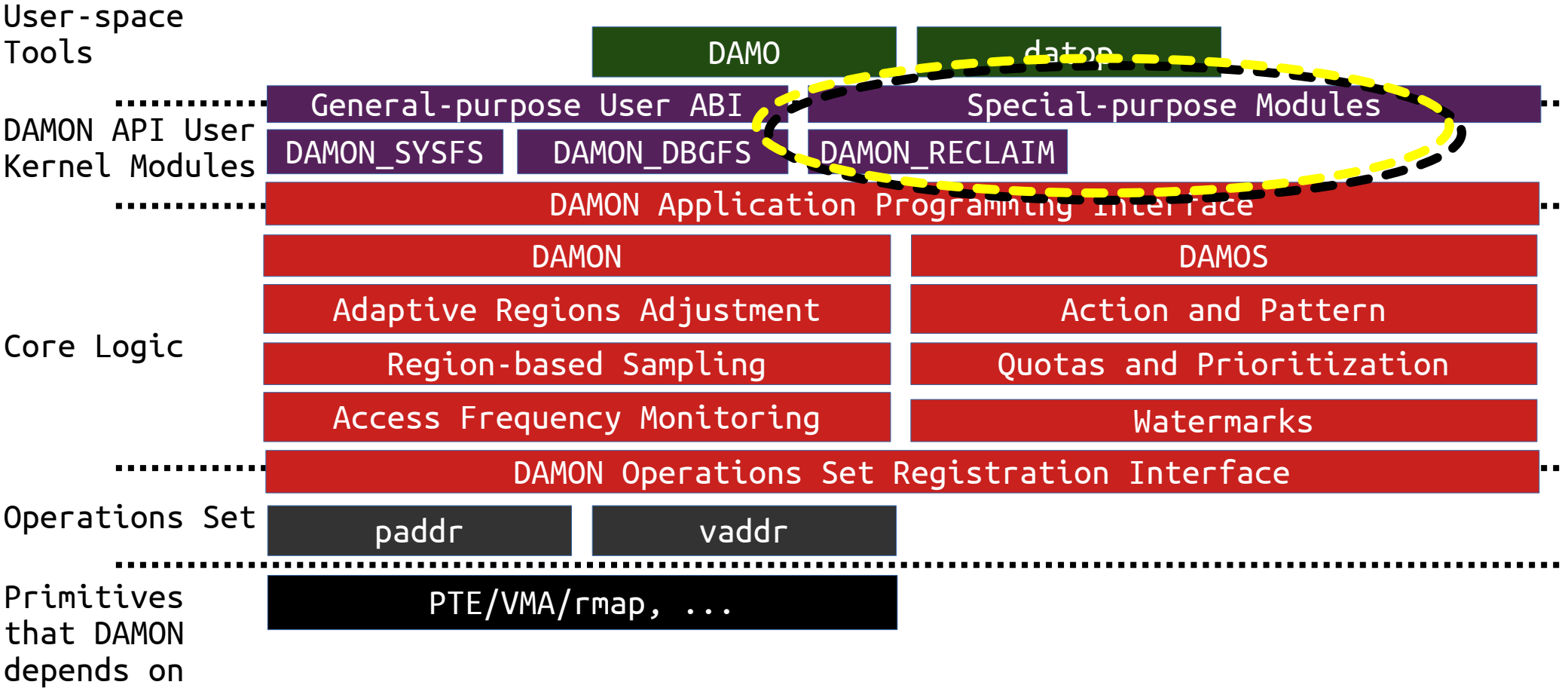
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



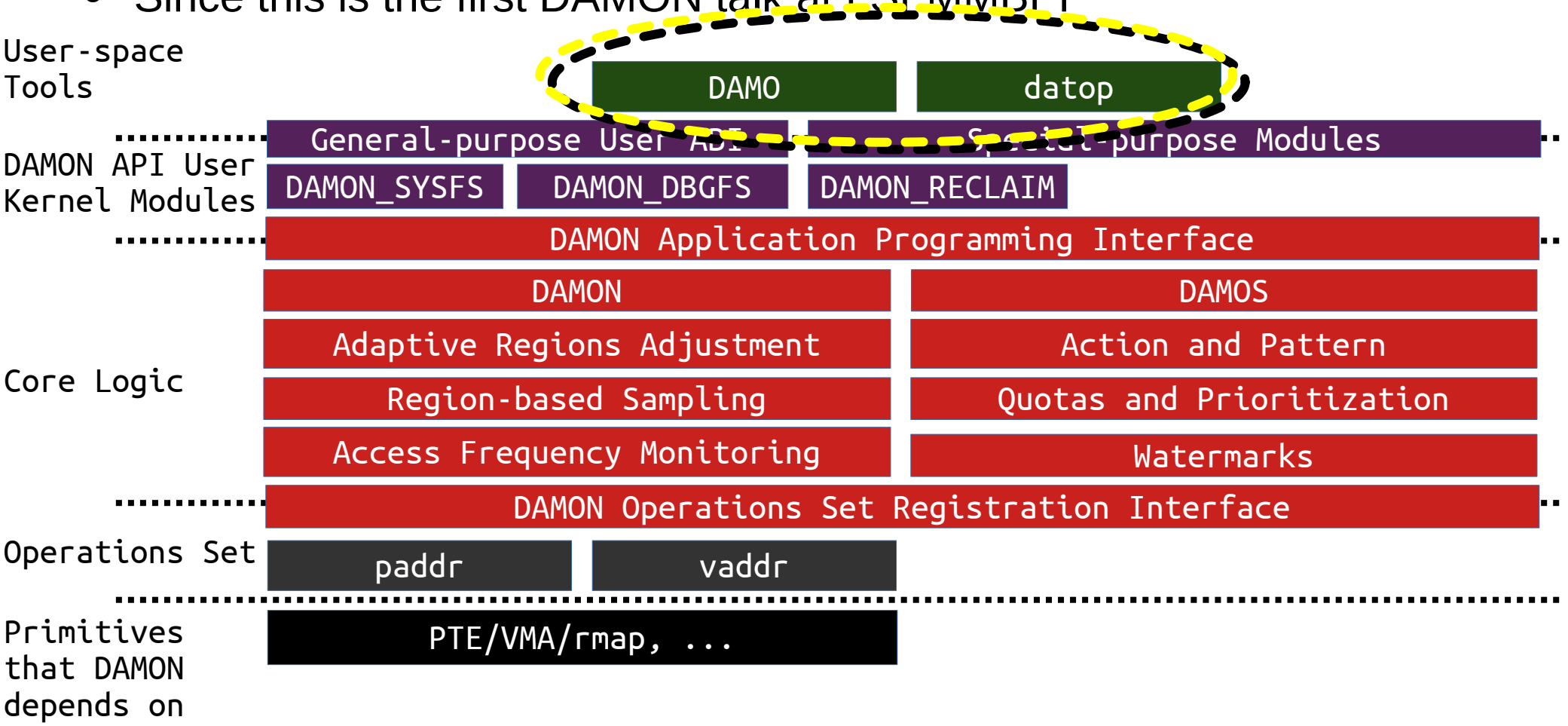
DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



DAMON Stack, as of 2022-05-04 (LSF/MM/BPF 2022)

- About one year and four days ago
- Since this is the first DAMON talk at LSFMMBPF



Recap Until LSF/MM/BPF 2022 (~2022-05-04, or v5.18)

- DAMON (v5.15)
  - Virtual address spaces
- DAMON_DBGFS (v5.15)
- DAMON_PADDR (v5.16)
- DAMOS (v5.16)
  - Action, Access pattern, Quotas, Watermarks
- DAMON_RECLAIM (v5.16)
- DAMON_SYSFS (v5.18)

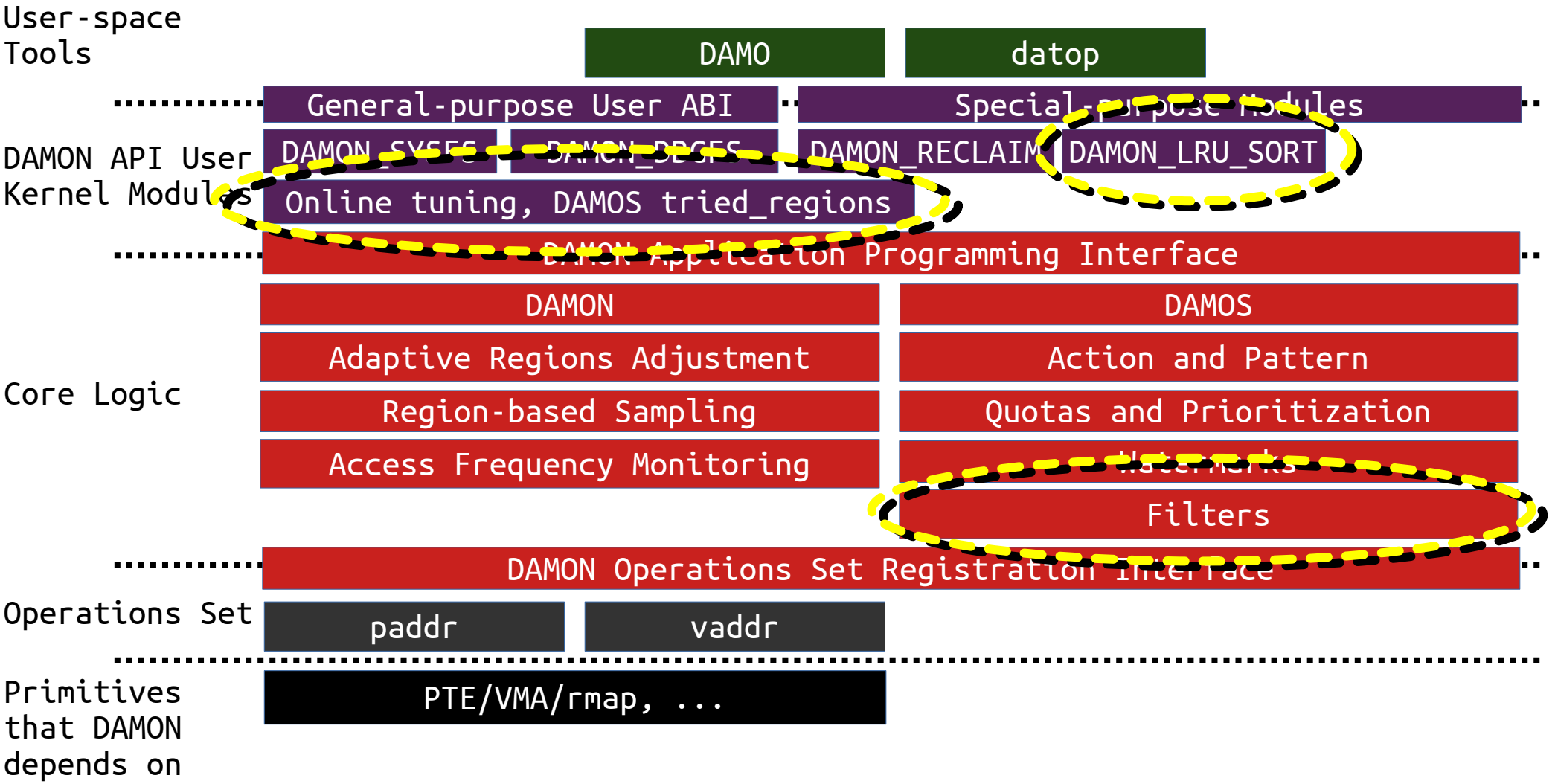
Updates of DAMON Development since LSFMM22 (1/2)

- Online Tuning (v5.19)
  - Allow DAMON_SYSFS and DAMON_RECLAIM users update parameters online (API users were able to do from the beginning)
  - Before this, DAMON should be restarted with new parameters
  - Makes user-space driven feedback-based auto-tuning available
- LRU_SORT (v6.0)
  - Mark hot memory region pages active and vice versa
  - Reduce memory PSI about 20%
  - Fine tuning is required

Updates of DAMON Development since LSFMM22 (2/2)

- Sysfs: Damos Tried Regions (v6.2)
  - Previously, only tracepoints were provided
    - Inefficient for getting only a single snapshot of specific pattern
  - Let Users Show DAMOS-found regions of specific pattern in detail
  - Allow query-like efficient monitoring results retrieval
- Damos Filters (v6.3)
  - Filter-in/out specific type of pages
  - Anonymous page and pages of specific cgroups are supported
  - Combinations of arbitrary number of filters are available
  - Can apply scheme to only anon pages, non-anon pages, only specific cgroups, all but except specific cgroups, and/or any combination

DAMON Stack, as of LSF/MM/BPF 2023 (2023-05-08)



Future Plans, Short Term (2023) 1/4

- Write-only Monitoring
  - Expect to be useful for
    - Writes limited flash-like devices
    - Live migration target selection
    - KSM usage was also suggested by the community
  - Add a new Operations Set using soft-dirty PTE as its primitive
  - A patchset was once posted, but postponed due to soft-dirty PTE issues
    - Is the issue still going on?

Future Plans, Short Term (2023) 2/4

- Feedback-based quota auto-tuning
  - Motivated by TMO and DAMOOS
  - Having optimal aggressiveness of given scheme is important
  - DAMOS has aggressiveness control mechanism: Quotas
  - Finding the optimum aggressiveness and translate to quotas is difficult
  - Idea: receive feedback about the aggressiveness and adaptively tune it
  - Pre-existing quotas will still be respected as hard limit quotas
  - Support DAMOS self-collectable metrics as being used as feedback
    - e.g., PSI, # page faults, etc
    - Results in nearly self-tuning schemes (kernel that just works?)

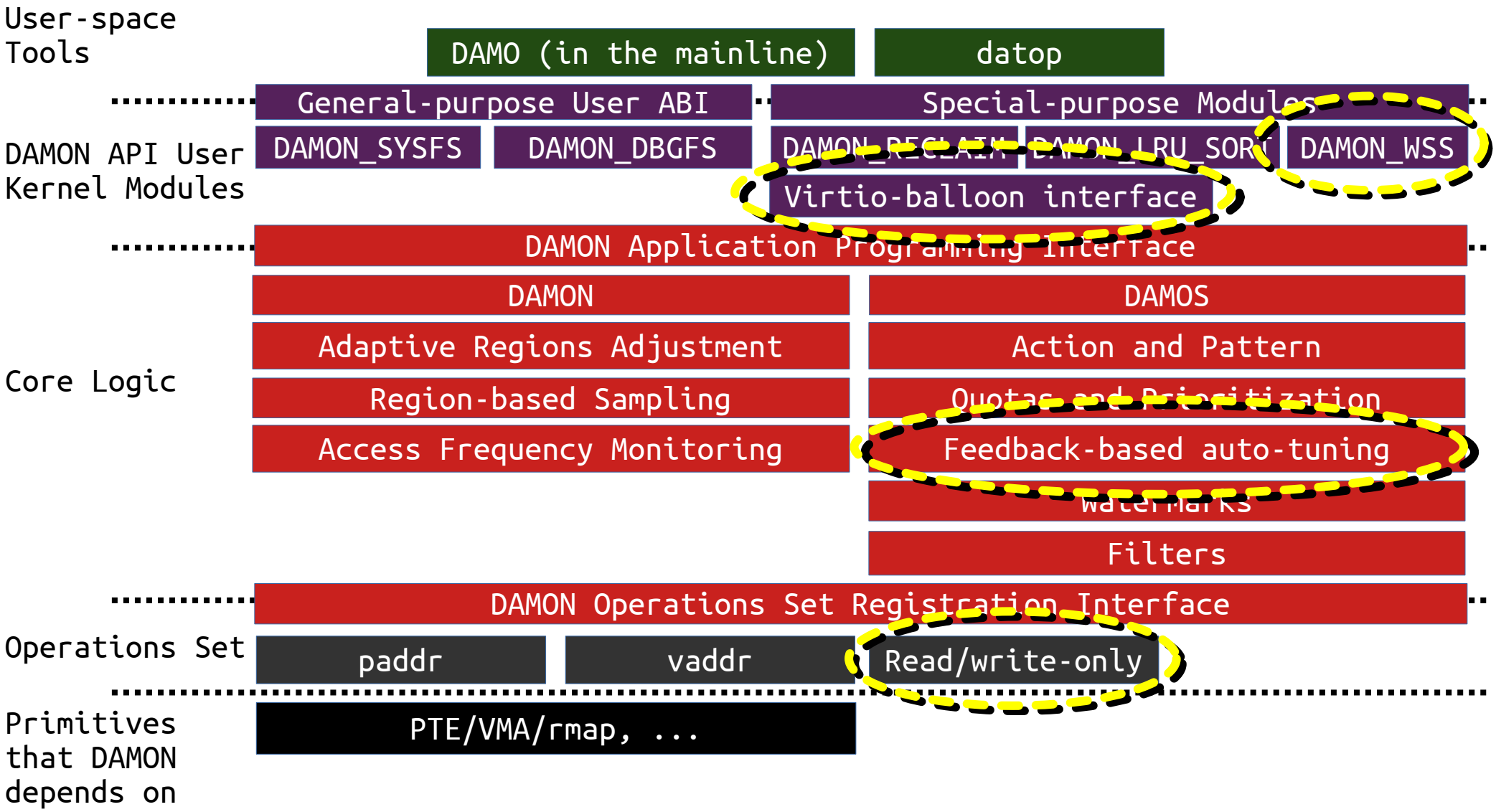
Future Plans, Short Term (2023) 3/4

- DAMON_RECLAIM control via virtio-balloon with Free Pages Reporting
  - One expected DAMON_RECLAIM usage is running it with PAGE_REPORTING in guest VMs on memory-overcommitted host
  - Currently, guest users should voluntarily enable/tune DAMON_RECLAIM
  - Applying rules/interfaces similar to PAGE_REPORTING may be better
  - Could there be some new interface for such usecases in general?
- In-tree DAMON User-space tool
  - DAMON User-space tool is out-of-tree, at Github
  - Ingesting it in the tree may help
    - Users understanding the ABI usage
    - Avoid ABI-tooling being broken
  - Out-of-tree version supports all kernels; should in-tree version different?

Future Plans, Short Term (2023) 4/4

- DAMON_WSS
  - Yet another DAMON module for simple working set size retrieval
  - Can be useful for optimal memory size estimation
  - Already available with DAMON user-space tool, but simpler interface is preferred
  - Using its module parameter would be straightforward, but would it make sense to add the output to ‘/proc/meminfo’ under explicit config option?

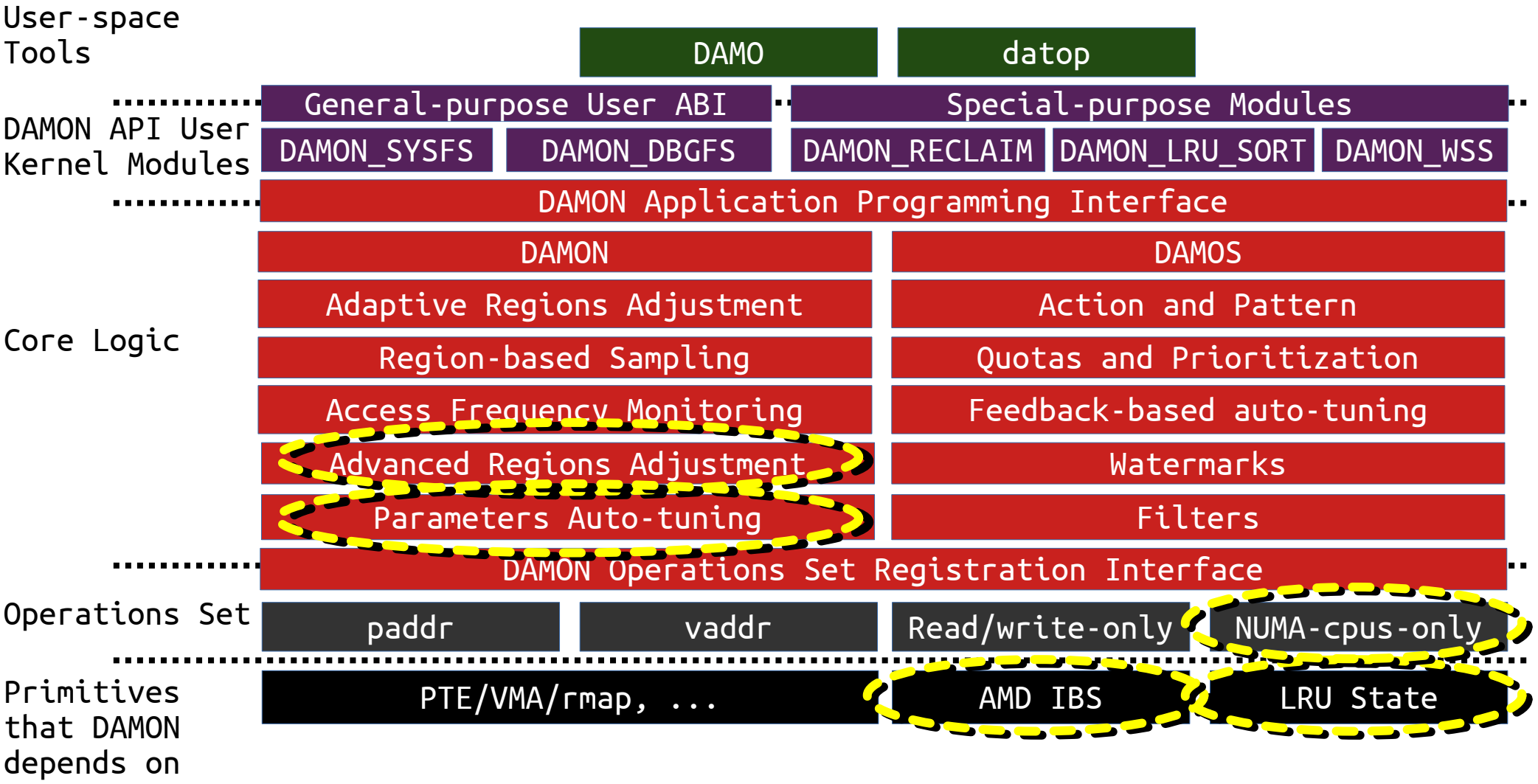
DAMON Stack, hopefully by end of 2023



Future Plans, Long Term (Not yet scoped)

- More DAMON Operations Sets
  - Page granularity support (LRU as underlying primitive)
  - Specific CPUs-only access monitoring
  - IBS-like H/W access check feature as underlying primitive
- More DAMOS Actions and DAMOS Modules
  - Tiered memory management
    - SJ don’t have good tiered memory development environment, any easy way?
  - THP memory footprint reduction
  - NUMA balancing
- DAMON Improvement
  - Accuracy
  - Tuning

DAMON Stack, In a Future



Questions and/or Comments, Please

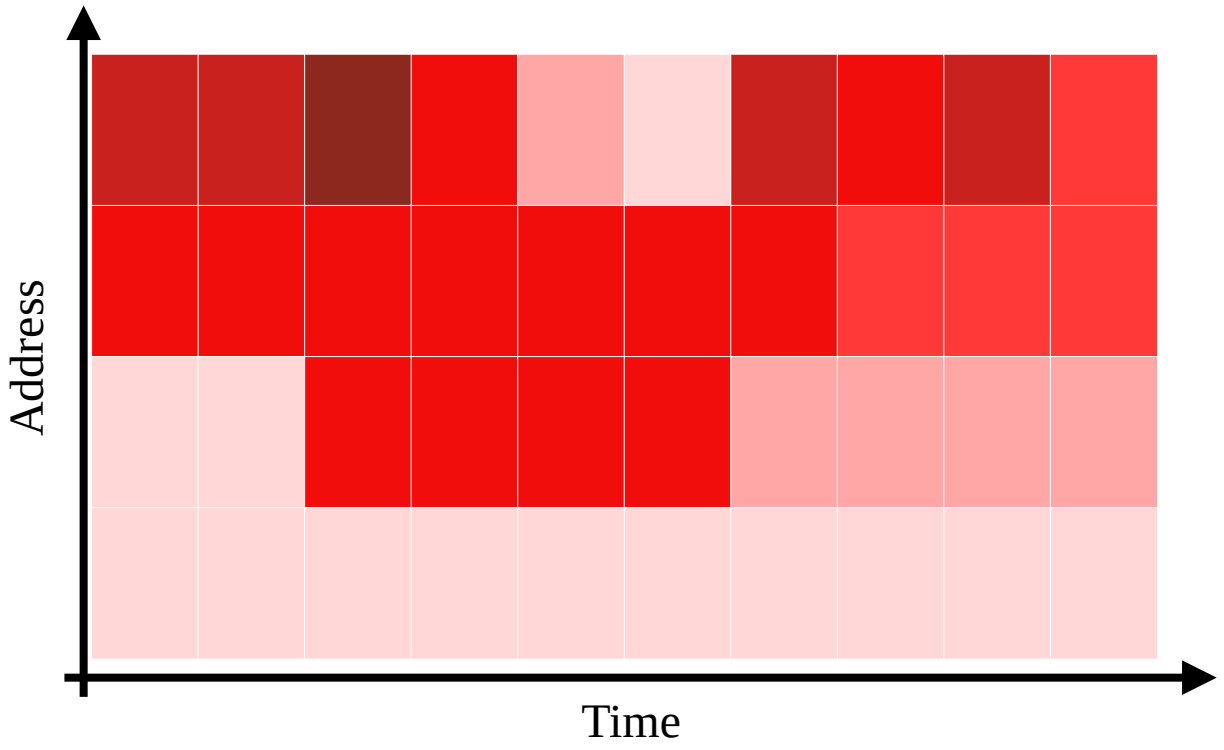
- For example,
  - I want this plan to be [de]prioritized
  - I'd suggest adding this idea to the plan
  - I think this problem can be work-arounded by…
  - Run this benchmark for that plan
- You can also use below
  - Project page: **https://damonitor.github.io**
  - Kernel docs for **admin** and **programmers**
  - DAMON mailing list: **damon@lists.linux.dev**
  - DAMON Beer/Coffee/Tea **Chat**
  - The maintainer: **sj@kernel.org**

https://kids.nationalgeographic.com/content/dam/kids/photos/animals/Birds/A-G/adelie-penguin-jumping-ocean.ngsversion.1396530997321.adapt.1900.1.jpg

Backup Slides

DAMON: What It Provides?

- Conceptually, DAMON does periodic access check
  - Let users accumulated access checks results
- Allows users to know which memory area is how frequently accessed

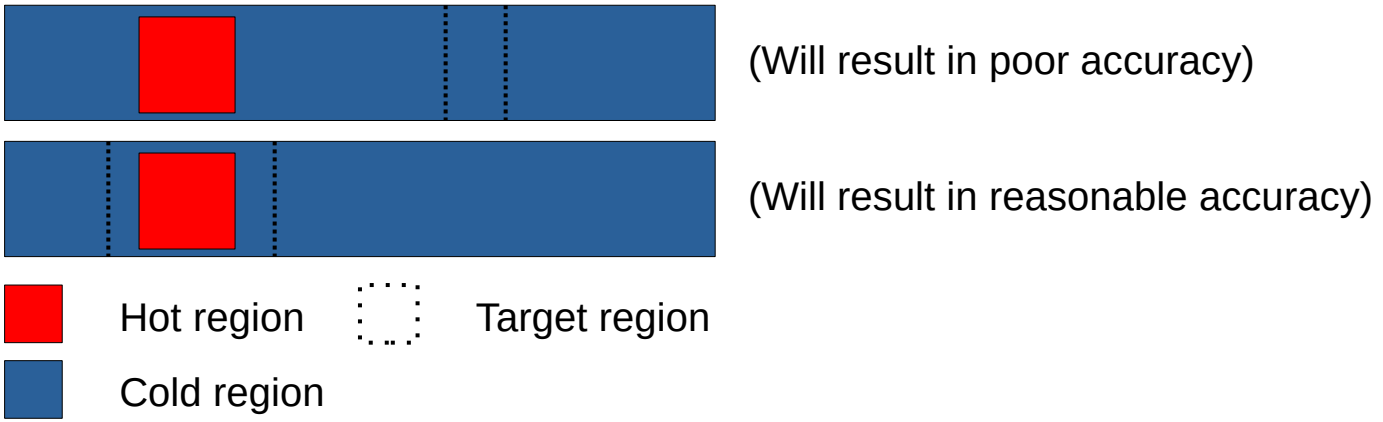


Conceptual Psudo-code of DAMON

```
while monitoring_on:
    for page in monitoring_target:
        if accessed(page):
            nr_accesses[page] += 1
    if time() % aggregation_interval == 0:
        for callback in user_registered_callbacks:
            callback(monitoring_target, nr_accesses)
        for page in monitoring_target:
            nr_accesses[page] = 0
    sleep(sampling interval)
```

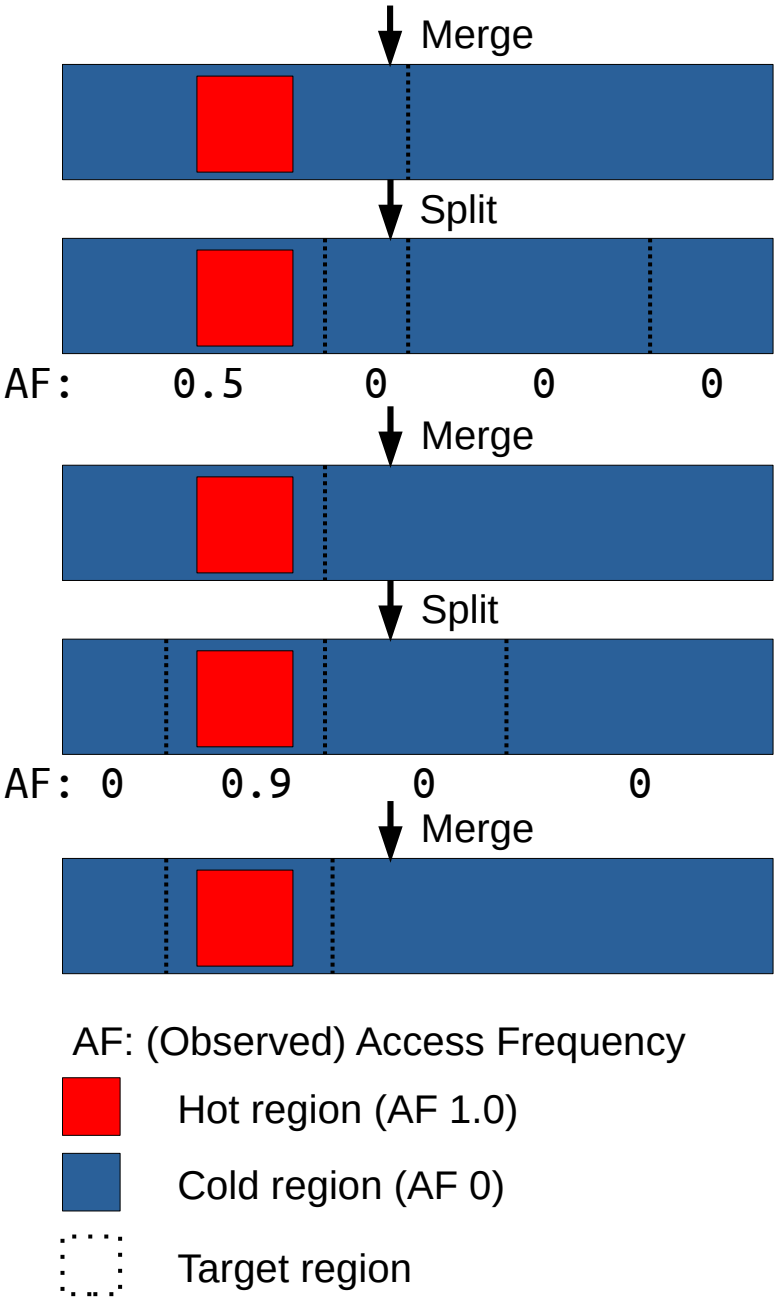
Region-based Sampling

- Defines data objects in access pattern oriented way
  - “A data object is a contiguous memory region that all page frames in the region have similar access frequencies”
  - By the definition, if a page in a region is accessed, other pages of the region has probably accessed, and vice versa
  - Thus, checks for the other pages can be skipped
- By limiting the number of regions, we can control the monitoring overhead regardless of the target size
- However, the accuracy will degrade if the regions are not properly set



Adaptive Regions Adjustment

- Starts with minimum number of regions covering entire target memory areas
- For each aggregation interval,
  - merges adjacent regions having similar access frequencies to one region
  - Splits each region into two (or three, depend on state) randomly sized smaller regions
  - Avoid merge/split if the number of regions might be out of the user-defined range
- If a split was meaningless, next merge process will revert it (vice versa)
- In this way, we can let users control the upper bound overhead while preserving minimum and best-effort accuracy



DAMON User Interfaces: How You Can Use DAMON

- DAMON provides only kernel API for other kernel components
- There is a Linux kernel module named DAMON sysfs interface
  - Implement pseudo-files on sysfs
  - Control DAMON using DAMON API, based on I/O to the sysfs file
  - User-space users can control DAMON via the sysfs files
  - Manual use of the files is tedious, though
  - User-space tools doing the file operations instead can be developed

```
# cd /sys/kernel/mm/damon/admin/
# echo 1 > kdamonds/nr_kdamonds && echo 1 > kdamonds/0/contexts/nr_contexts
# echo vaddr > kdamonds/0/contexts/0/operations
# echo 1 > kdamonds/0/contexts/0/targets/nr_targets
# echo $(pidof <workload>) > kdamonds/0/contexts/0/targets/0/pid_target
# echo on > kdamonds/0/state
```

Example usage of DAMON sysfs interface

DAMOS for Access-aware Optimizations with No Code

- DAMOS is a feature of DAMON for offloading the effort to DAMON
  - Users can simply
    - specify the access pattern of their interest, and
    - the action they want to apply to the regions of the pattern
  - Then, DAMON finds regions of the pattern and apply the action
  - No code, just request specification
  - Provides some more important features, but out of scope of this talk

```
{
        "access_pattern": {
                "sz_bytes": {"min": "4K", "max": "max"},
                "nr_accesses": {"min": "0 %", "max": "0 %"},
                "age": {"min": "2 m", "max": "max"}
        },
        "action": "pageout"
}
```

A json-format DAMOS scheme asking
“Page out memory regions of >=4K that not accessed at all for >=2 minutes”