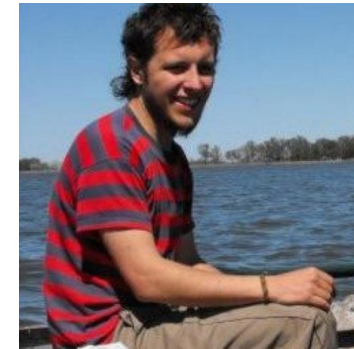


Bloc for Pharo

Modern Foundations

Martín Dias & Pablo Tesone | ESUG 2026, Plovdiv



Martín Dias



Pablo Tesone



About the Presenters



Martín Dias

- Pharo Consortium Engineer
- Email: tinchodias@gmail.com
- GitHub: @tinchodias



Pablo Tesone

- Pharo Consortium Engineer
- Email: pablo.tesone@inria.fr
- GitHub: @tesonep



Bloc is made of many pieces...

- **Bloc**: the core graphic framework
- **Toplo**: the widget set
- **Spec-Toplo**: the integration between Toplo & Spec
- **Alexandrie**: the bindings with Cairo & low level operations
- **Album**: Our text editor component



**Just a
Reminder**

**Note: Commits are calculated exclusively from the remote-only origin/dev branch since ESUG 2025.*



Annual Development Activity

1,378

Commits on origin/dev
branch this year
(July 2025 - June 2026)

- **Toplo**: 886 commits (Lists unification, tables/trees, circular menus, diffs)
- **Spec-Toplo**: 192 commits (Spec2 vector adapter coverage)
- **Bloc**: 144 commits (OSWindow-SDL3 host, BAnchorLayout, BIBounds)
- **Alexandrie**: 79 commits (Cairo vector rendering engine updates)
- **Album**: 77 commits (CI fixed, text model updates, ALL TESTS GREEN!)

**Note: Commits are calculated exclusively from the remote-only origin/dev branch since ESUG 2025.*



Bloc Core Upgrades (I)

- Ignored Layouts: Unified anchoring for elements bypass layout flow rules
- Geometry Coordinates: Migrated boundary constraints from Rectangle to B1Bounds
- Dynamic Transforms: Native support for parent-relative scale and rotate animations
- API Alignment: Large-scale transition from #size: to #extent: across layout hierarchies





Bloc Core Upgrades (II)

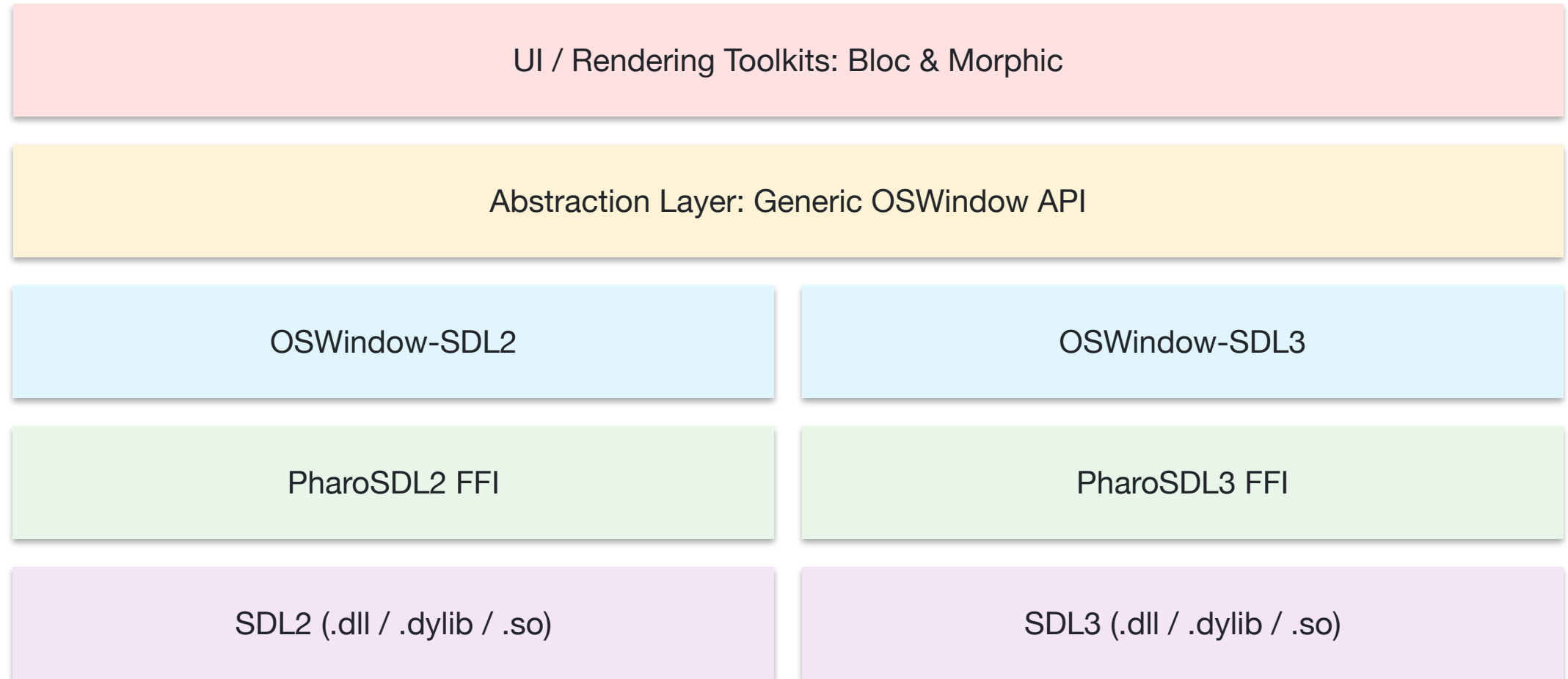
- Clean API: Deprecated `#labeled:` in favor of elements `#id:` and `#indexed:`
- Housekeeping: Deleted all deprecated methods older than 2 years (across Bloc, Album, and Toplo)
- Life-Cycle: Added `BLElementToBeRemovedFromParentEvent` and visibility layout reflows
- Traversal: Optimised depth-first and breadth-first element traversal algorithms
- Future compatibility: Semaphore waits adapted for Pharo 12, 13, and 14

**Getting
Better...**

Bloc & SDL3

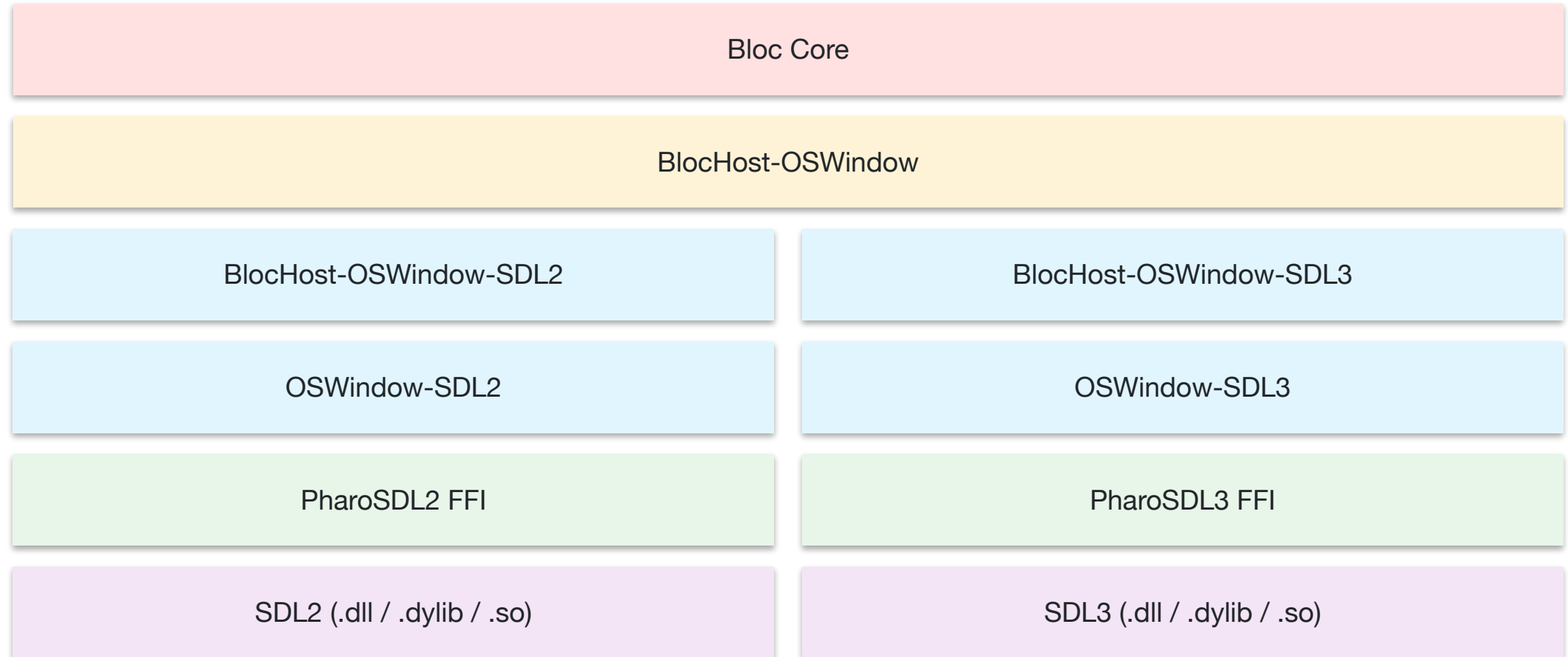


OSWindow Architecture Layers





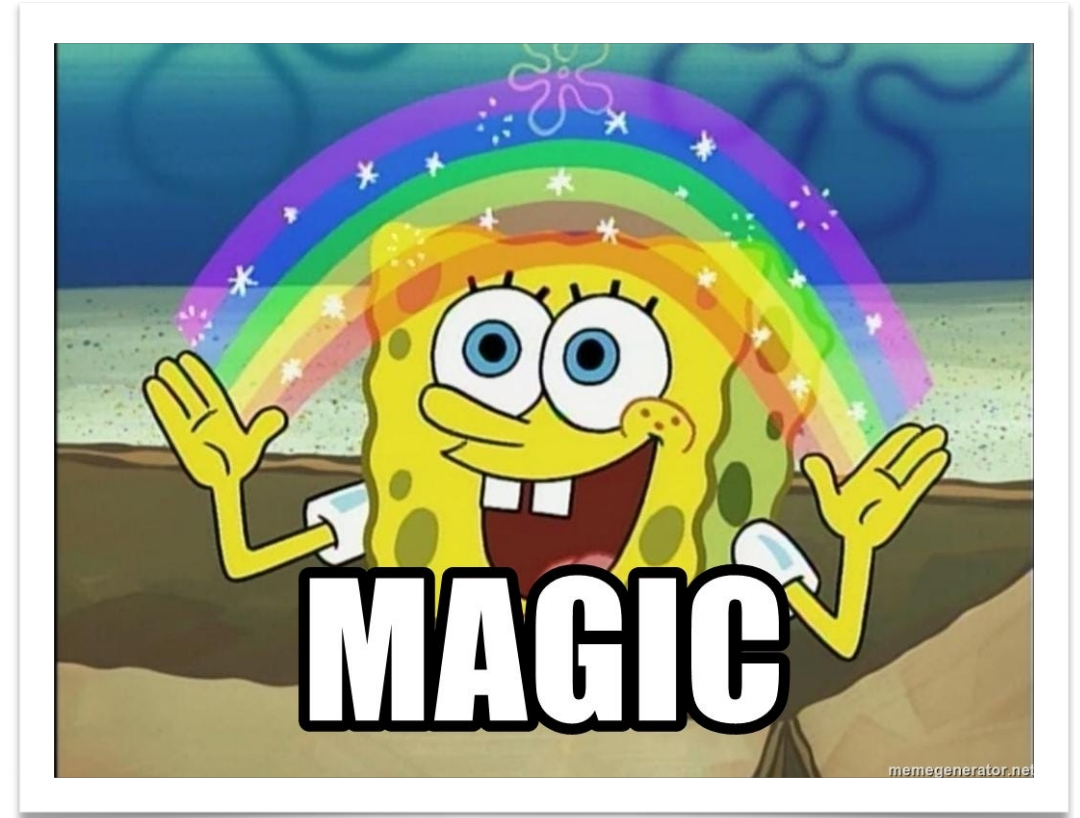
Bloc Architecture Layers





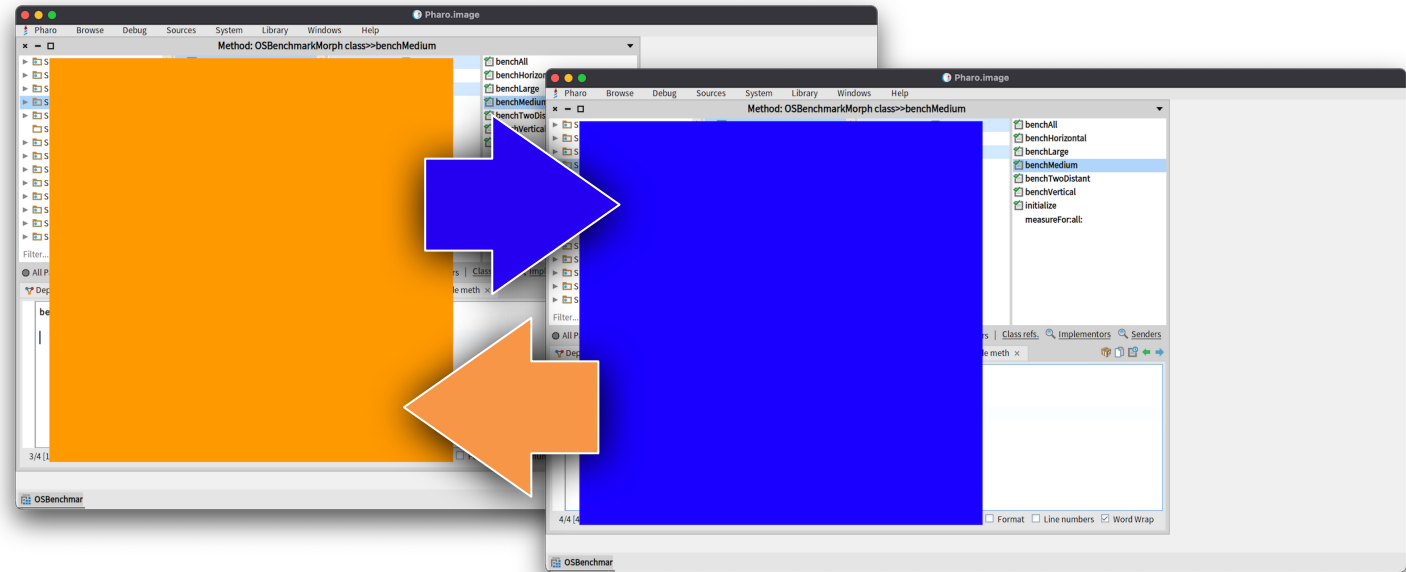
Transparent Solution

- Pharo VM includes the SDL3 Library
- During startup automatically selects SDL3 or SDL2 to use
- All integration is transparent to the user of OSWindows
- It is transparent for Morphic & Bloc



Screen Tearing or Process Starvation

Experiment:



UI Process (Priority 40)

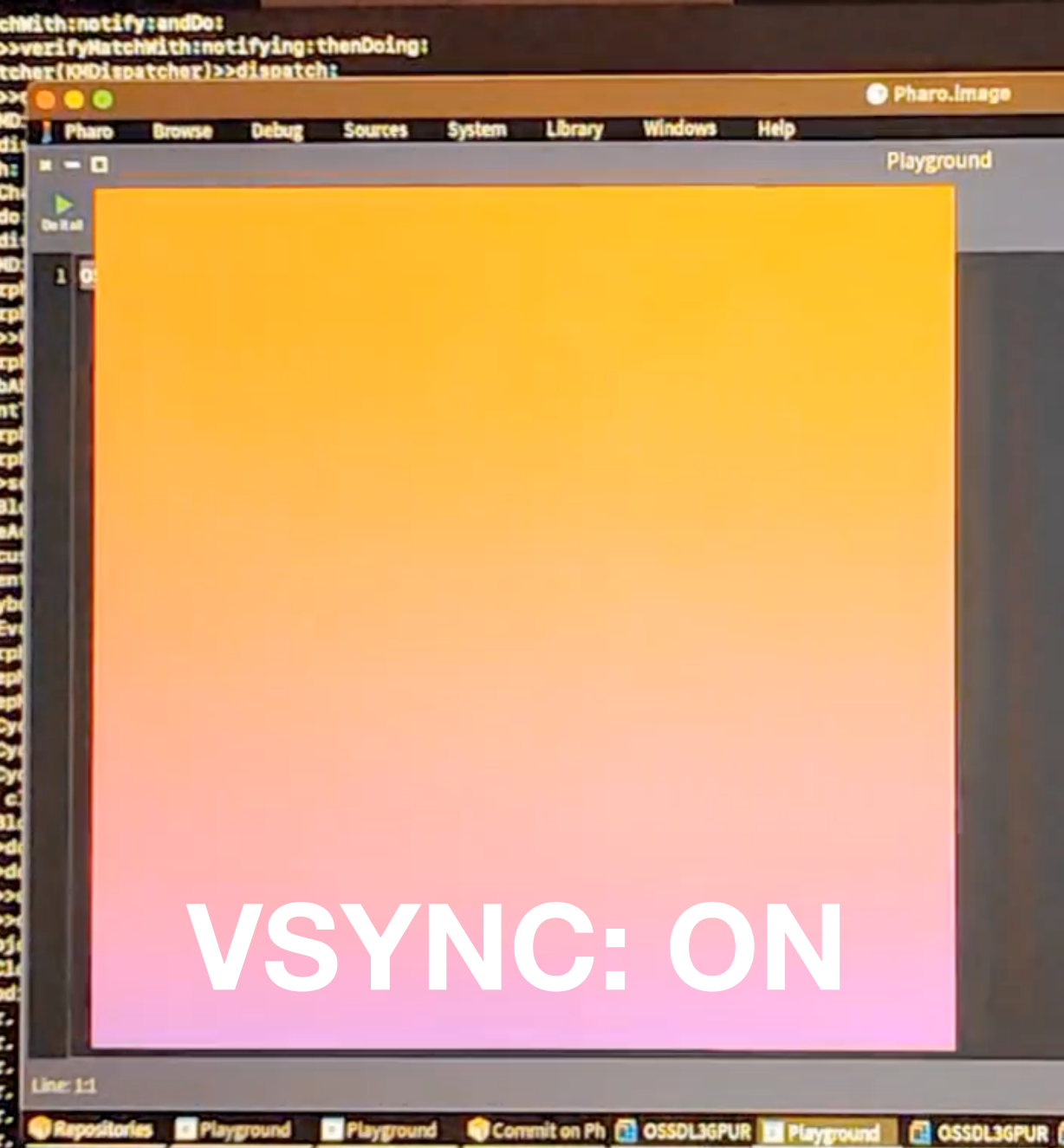
a OSBenchmarkMorph

```
stepCount := stepCount + 1.
```

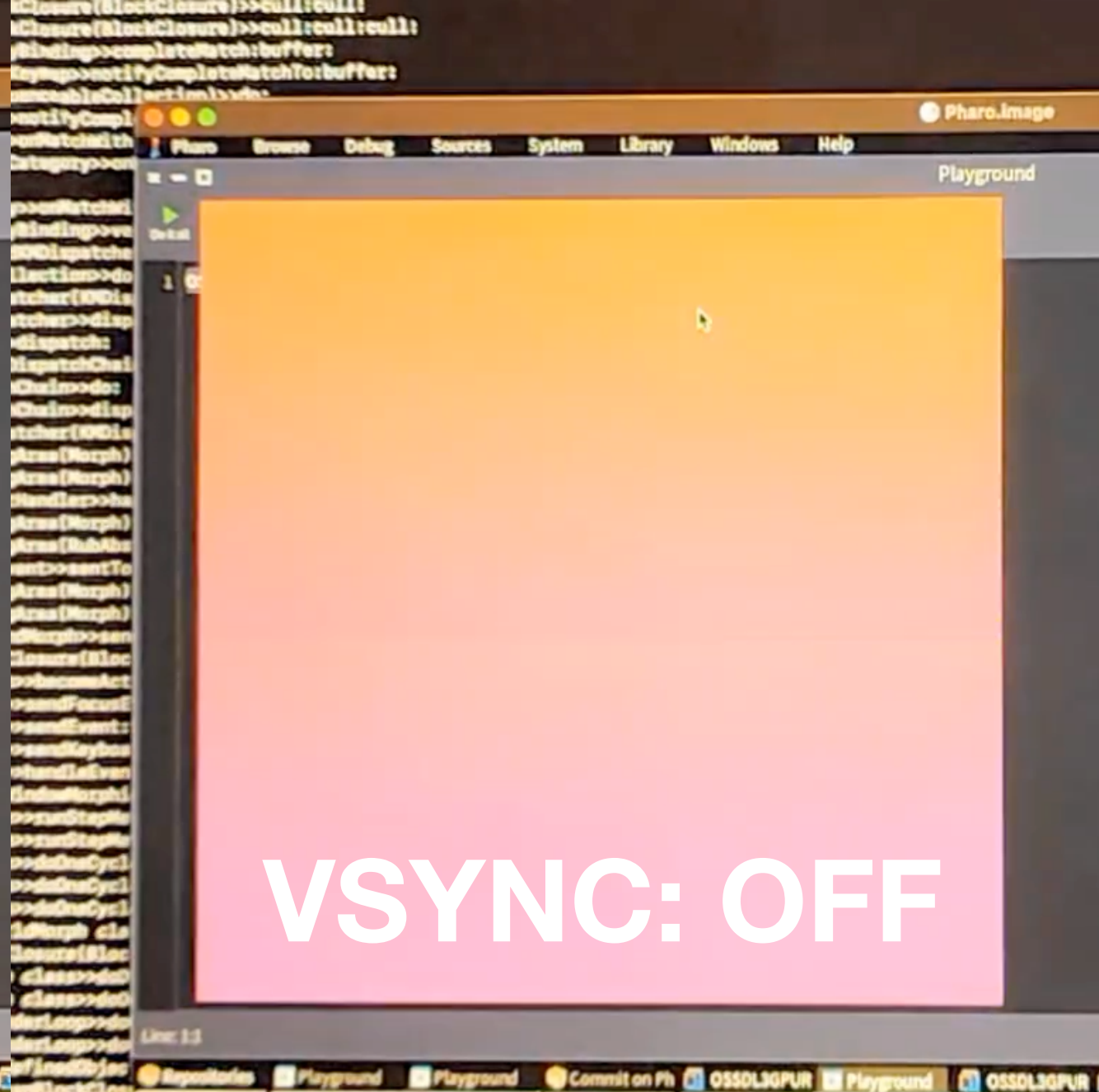
```
self color:  
  (stepCount even  
   ifTrue: [ Color orange ]  
   ifFalse: [ Color blue ])
```

Background Process (Priority 30)

```
backgroundCount := 0.  
countBackgroundProcess := [  
  [ backgroundCount := backgroundCount + 1 ] repeat  
] forkAt: 30.
```



VSYNC: ON



VSYNC: OFF

```
diess$ PHARO_WINDOW_DRIVER=OSSDL3Driver ./pharo-ui Pharo.image
diess$ ph
diess$ PHARO_WINDOW_DRIVER=OSSDL3Driver ./pharo-ui Pharo.image
diess$ ph
diess$ PHARO_WINDOW_DRIVER=OSSDL3Driver ./pharo-ui Pharo.image
diess$ ph
```

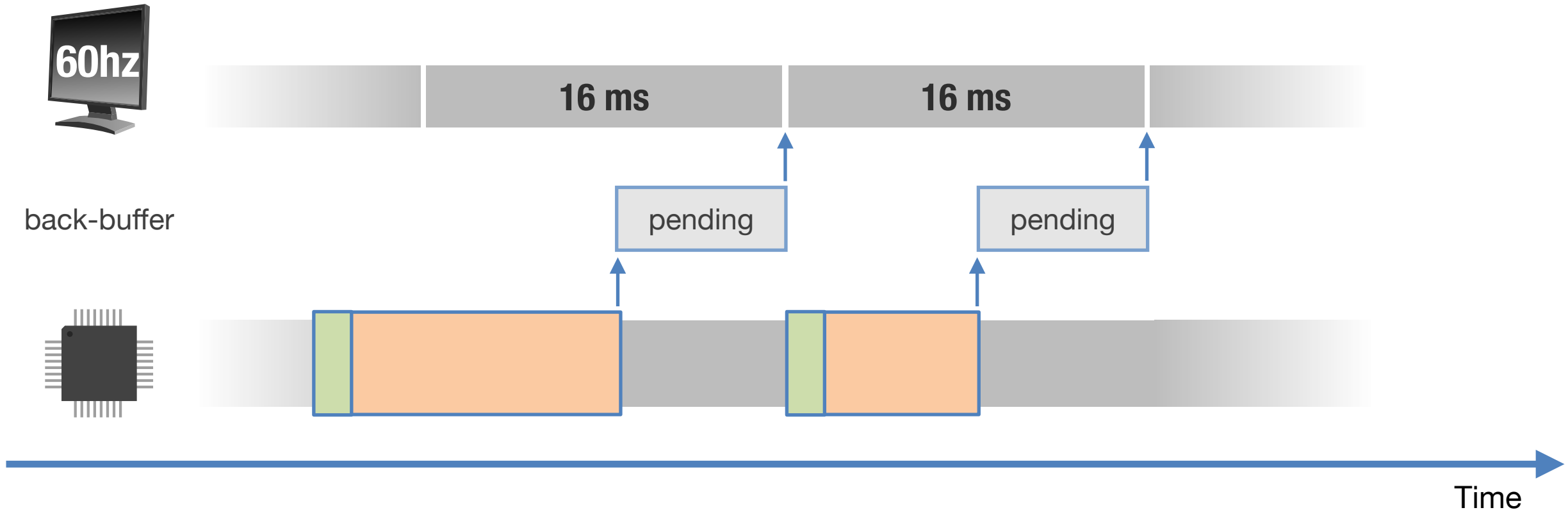
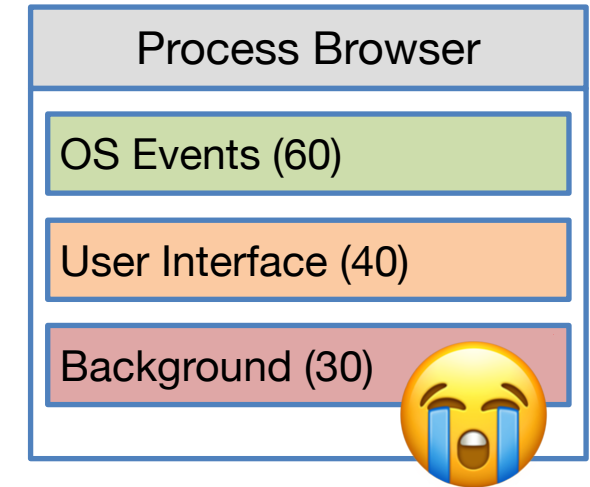
**screen
tearing**

old frame

new frame

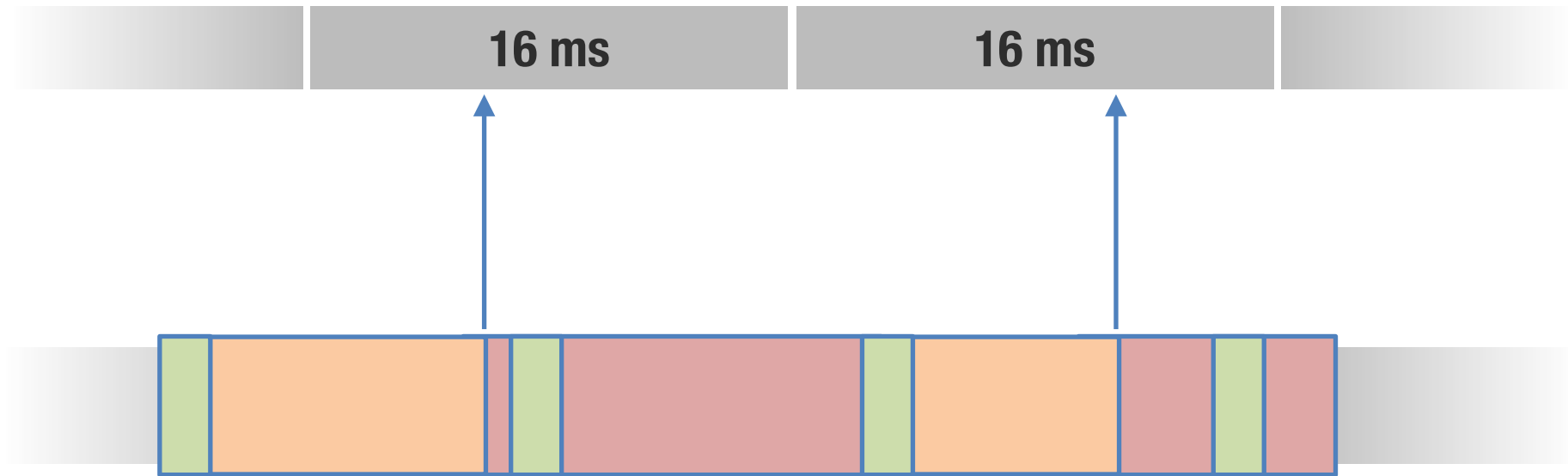
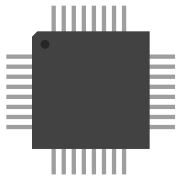
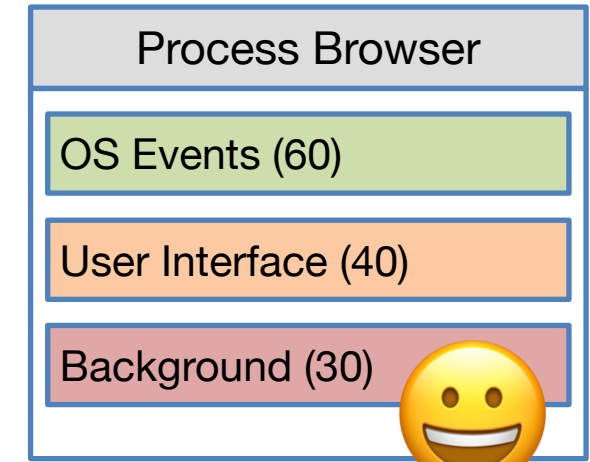
VSYNC: ON

OSSDL2Driver & OSSDL3Driver



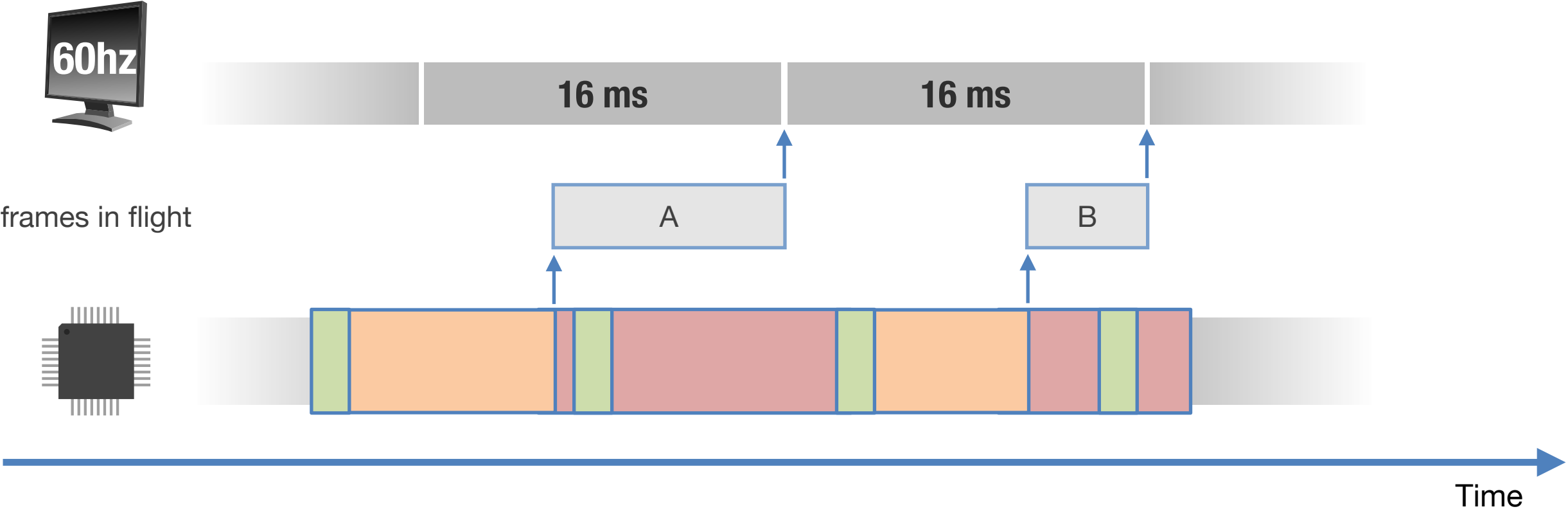
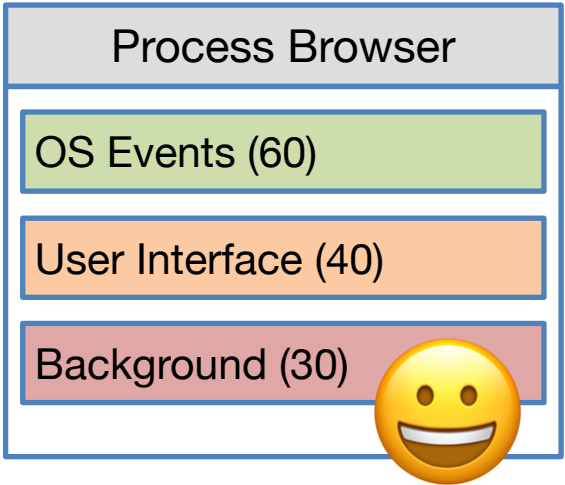
VSYNC: OFF

OSSDL2Driver & OSSDL3Driver



SDL3 GPU API

OSSDL3GPUDriver



Settings Browser



Load



Export

OSWindow-SDL3



OSWindow-SDL3

OSWindow-SDL3

Debug



Event poll period (ms)

5



GPU Swapchain Present Mode

Mailbox



Renderer VSync Mode

Enabled



Concurrency & Roadmap



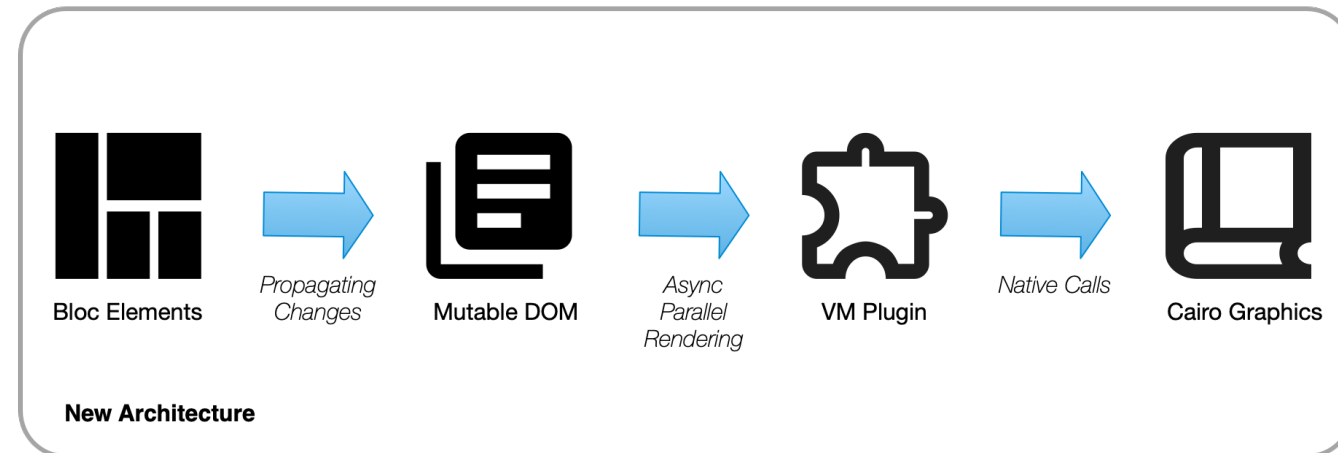
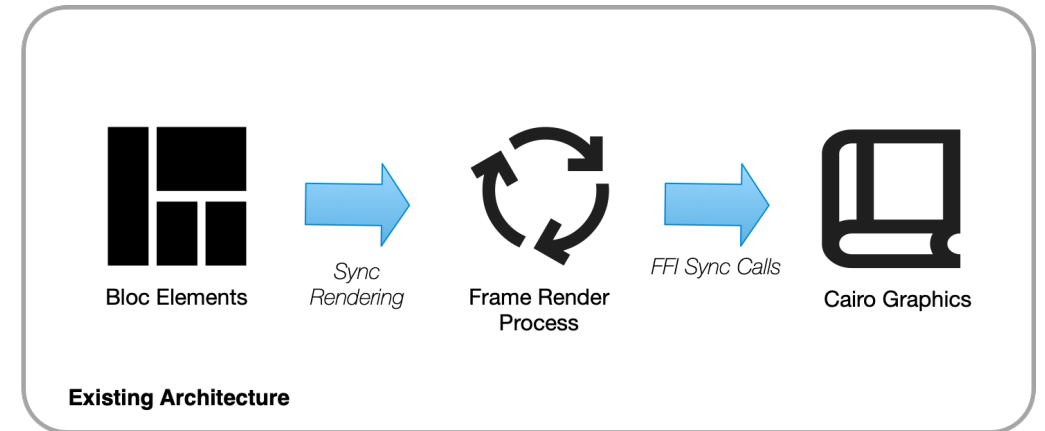
Single-Threaded Bottlenecks

- Pharo code runs in a single OS thread
- Cairo Contention: Athens and Cairo rasterisation drawing runs on this same single OS thread
- CPU Starvation: Blocking render cycles limit the execution time available for Pharo processes
- The Solution: Offload drawing workloads to separate threads/processes to keep the UI responsive



First Step: Mutable DOM & Render Plugin

- Adding a Mutable DOM
 - Registering changes
 - Independent of BElements
- Having an Async Rendering
 - Implemented as a Plugin
 - Reduce Marshallings
 - Render in separate OS thread





In the meantime: How we validate it

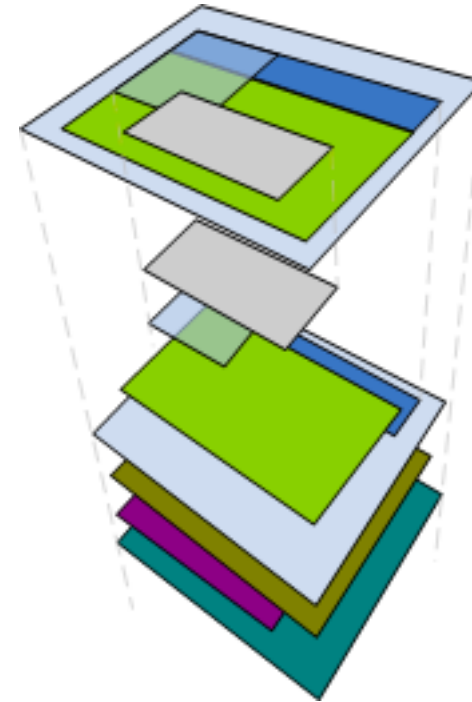
- Adding a lot of test
- Regression tests comparing the generated bitmaps
- Using mutation testing to find differences and performance issues (check Fede's IWST talk)

**+200 regression tests
+30 benchmarks &
stress test**



Next Steps

- Better handling of Layering
- Concurrent Layering rendering
- Composition using SDL3 + GPU
- Separating Application Logic and Rendering in different processes





The future...

- Concurrent rendering engine prototype & scaling tests in Bloc
- Validation of concurrent rendering architecture with Bloc components and Toplo widgets
- Spec2 library upgrades and support for new Toplo widgets (Accordion, Lists, and Trees)
- Album text editor evaluation to support inline visual objects, visual anchors, and Markdown editing
- Pharo World in Toplo/Bloc (Check tomorrows presentation)

**We will get
slowly...**



Thanks to all contributors!!

- Alain Plantec: Led the Toplo widget library redesign, unified lists/tables, and circular menus
- Benoit Verhaeghe: Developed the Spec-Toplo bridge and tabbed notebook adapters
- Stéphane Ducasse: Provided core layout alignment feedback, code reviews, and API alignments
- Martín Dias & Pablo Tesone: Core VM bindings, FFI wrappers, and SDL3 driver integration
- Enzo Demeulenaere, Cyril Ferlicot & Kris: Added test suites, baselines, and CI integrations



**STAY
TUNED!**



- Mailing-list: lse-openbloc@inria.fr (subscribe at: <http://sympa.inria.fr>)
- Discord: Pharo server / #bloc channel
- **Bloc:** <https://github.com/pharo-graphics/Bloc>
- **Toplo:** <https://github.com/pharo-graphics/Toplo>
- **Spec-Toplo:** <https://github.com/pharo-graphics/Spec-Toplo>
- Next Session: Join us tomorrow for Spec-Toplo!

**Check
Released
Versions!!!**