

D3 Migration Audit and Plan

Date: 2026-01-19 **Status:** Active **Goal:** Migrate all deployed code to native zoom/drag/transitions, then eliminate d3-selection

Executive Summary

This audit identifies all built/bundled code in the PSD3 monorepo, their D3 dependencies, and migration requirements. The goal is to:

1. Migrate all active code to use native implementations (Pointer Events zoom/drag, pure PS transitions)
 2. Do a clean build, deploy, and test on MacMini
 3. Remove remaining d3-selection dependencies where feasible
-

Deployed Services (docker-compose.yml)

Tier 1: Core Infrastructure

Service	Context	D3 Usage	Migration Status
edge	scuppered-ligature	None (Lua/Nginx)	N/A - no migration needed
website	site/website	Heavy - all D3 via PSD3 wrappers	Primary target

Tier 2: Full-Stack Showcases

Service	Context	D3 Usage	Migration Status
tidal-backend	psd3-tilted-radio/purerl-tidal	None (Erlang)	N/A
tidal-frontend	psd3-tilted-radio/purescript-psd3-tidal	Minimal via PSD3	Low effort
ee-backend	hypo-punter/ee-server	None (Python)	N/A
ee-frontend	hypo-punter/ee-website	Heavy direct D3	High effort
ge-backend	hypo-punter/ge-server	None (Python)	N/A
ge-frontend	hypo-punter/ge-website	Heavy direct D3	High effort
ce-backend	corrode-expel/ce-server	None (Node.js)	N/A
ce-frontend	corrode-expel/ce2-website	Moderate D3	Medium effort
sankey	psd3-arid-keystone	D3 sankey only	Low effort
wasm-demo	wasm-force-demo	Minimal - WASM handles force	Low effort

Tier 3: PureScript Showcases

Service	Context	D3 Usage	Migration Status
optics	emptier-coinage	Via PSD3	Should use new patterns
zoo	psd3-prim-zoo-mosh	Via PSD3	Should use new patterns
honeycomb	psd3-honeycomb	Via PSD3	Should use new patterns
anscombe	psd3-anscombe-quartet	Via PSD3	Should use new patterns
layouts	allergy-outlay	Via PSD3	Should use new patterns

Tier 4: Library Landing Pages

Service	Context	D3 Usage	Migration Status
lib-selection	site/lib-selection	Via PSD3	Should use new patterns
lib-simulation	site/lib-simulation	Via PSD3	Should use new patterns
lib-layout	site/lib-layout	Via PSD3	Should use new patterns
lib-graph	site/lib-graph	Via PSD3	Should use new patterns
lib-music	site/lib-music	Via PSD3	Should use new patterns

Not Deployed (can skip migration)

Directory	Status	Action
graph-algos	Not in docker-compose	Skip
psd3-lorenz-attractor	Not in docker-compose	Skip
psd3-timber-lieder	Not in docker-compose	Skip
psd3-topics	Not in docker-compose	Skip
purescript-makefile-parser	Build tool only	Skip

Website Demo Pages - D3 Usage Analysis

Based on `Types.purs` Route definitions, key pages requiring attention:

Already Migrated / Using New Patterns

- `SimpleForceGraph` - Using native zoom/drag via Behavior types
- `TourMotionScrolly` - Migrated to `withPureTransitions`
- `PieDonutDemo` - Pure PS arc generation
- `AnimatedAttrTest` - Pure PS transitions
- `GUPAnimatedTest` - Pure PS with AnimatedAttr

Needs Verification

Route	Component	Likely Status
ForcePlayground	ForcePlayground.purs	Uses Behavior types - should work
TreeBuilder*	TreeBuilder/*.purs	Uses Behavior.Zoom - should work
SPLOM	SPLOM.purs + SPLOM.js	Has FFI - needs audit
LesMis*	Various	Uses force + GUP - verify patterns
TourHierarchies	Tour pages	Tree layouts - verify
TourFlow	Chord/Sankey	Verify native patterns

High-Priority FFI Files

These JS files have direct D3 usage that needs migration:

1. `src/Viz/SPLOM/SPLOM.js` - d3-selection for point visibility
2. `src/TreeBuilder/App.js` - d3-sankey for watermark
3. `src/Component/SankeyDebug/FFI.js` - d3-sankey direct

External Showcases - Detailed Analysis

hypo-punter (EE + GE) - HIGH EFFORT

`ee-website/src/Viz/UMAPScatter.js`:

- Uses `window.d3` global
- `d3.zoom()` with custom filter
- `d3.brush()` for selection
- `d3.scale*()` for axes

`ee-website/src/Viz/SPLOM.js`:

- Similar to UMAP
- Multi-panel brush coordination
- Zoom with scale rescaling

`ge-website/src/GE/Viz/NetworkGraph.js`:

- `d3-zoom`, `d3-drag`, `d3-force` direct usage
- Full D3 simulation management
- Custom drag/zoom coordination

Migration Approach: These are standalone apps with their own bundling. Options:

1. Rewrite to use PSD3 (high effort)
2. Keep D3 for these isolated apps (pragmatic)
3. Partial migration - keep D3 force/scale, replace zoom/drag

corrode-expel (Code Explorer) - MEDIUM EFFORT

ce2-website/src/Viz/Triptych/*.js:

- d3-selection for hover sync
- Cross-panel coordination
- No zoom/drag

Migration: Replace `select/selectAll` with native DOM or thin wrapper

psd3-arid-keystone (Sankey) - LOW EFFORT

- Uses d3-sankey for layout algorithm only
- No interactive D3 (zoom/drag/brush)
- Keep d3-sankey, it's an algorithm not DOM manipulation

Migration Plan

Phase 1: Verify Website (2-3 hours)

1. **Build clean:** `make clean && make website`
2. **Test key pages locally:**
 - `/#/simple-force-graph` (zoom/drag)
 - `/#/tour/scrolly2` (transitions)
 - `/#/pie-donut-demo` (pure arc)
 - `/#/force-playground` (complex force)
 - `/#/tree-builder` (zoom)
 - `/#/splom` (brush - needs FFI check)
3. **Fix any issues found**

Phase 2: Build All Apps (1-2 hours)

```
make clean-deps  
make all  
make verify-bundles
```

Fix any build failures.

Phase 3: Deploy to MacMini (30 min)

```
/deploy all
```

Phase 4: Test on MacMini (1-2 hours)

Test each deployed service:

- / (website)
- /ee, /ge (hypo-punter)
- /sankey
- /code (corrode-expel)
- /tidal
- /wasm
- /psd3/* (lib sites)

Phase 5: Assess External Apps (decision point)

For hypo-punter and corrode-expel:

- **Option A:** Keep their D3 usage (isolated, different bundle)
- **Option B:** Migrate to PSD3 patterns (high effort)

Recommendation: Option A for now. These are separate codebases with their own package.json. They don't affect the psd3-selection library's D3 footprint.

Phase 6: Final d3-selection Audit

After all deployed code is verified working:

1. Audit remaining d3-selection usage in psd3-selection FFI
 2. Identify what can be replaced with native DOM
 3. Create migration plan for library-level d3-selection elimination
-

Current D3 Dependencies in psd3-selection

Required (keep):

- d3-brush - Complex interaction, no native replacement yet
- d3-force - Physics simulation (core feature)
- d3-interpolate - Color interpolation
- d3-scale - Scale functions (core feature)
- d3-scale-chromatic - Color schemes
- d3-selection - DOM manipulation (Phase 6 target)
- d3-shape (line only) - Line generator
- d3-transition - Still used in some paths

Eliminated:

- d3-drag → PSD3.Interaction.Pointer
 - d3-zoom → PSD3.Interaction.Zoom
 - d3-chord → PSD3.Layout.Chord
 - d3-hierarchy → PSD3.Layout.Hierarchy.*
 - d3-ease → PSD3.Transition.Engine
 - d3-shape (arc) → PSD3.Internal.Generators.Arc
-

Success Criteria

1. All Docker services start and pass healthchecks
 2. Website demos work with native zoom/drag/transitions
 3. No JavaScript console errors on tested pages
 4. Performance is equivalent or better than before
 5. d3-dependencies.json accurately reflects actual usage
-

Appendix: Build Commands Quick Reference

```
# Clean everything  
make clean-deps  
  
# Build all  
make all  
  
# Build specific  
make website  
make apps  
make lib-sites  
  
# Deploy  
/deploy all  
  
# Status  
/deploy status
```