SubsystemEvidence

September 4, 2026

SubsystemEvidence

Kind: Interface

Source: `atloria-monorepo/libs/parser-core/src/techdocs/subsystem-narrative.ts`

What the parse knows about a subsystem beyond its size and its neighbours.

Properties

Property	Type
`types`	`Array<[string, number]>`
`fileCount`	`number`
`busiestFile`	`{ path: string; entities: number }`
`hub`	`{ name: string; internalDependents: number }`
`widest`	`{ name: string; methods: number }`
`reachedFromOutside`	`Array<{ name: string; edges: number }>`
`surface`	`{`

```
endpoints: number;
methods: Array<[string, number]>;
basePaths: string[];
controllers: Array<{ name: string; endpoints: number }>;
controllerCount: number;
/** Guards / middleware declared on the endpoints, most-used first. */
guards: Array<[string, number]>;
/** Endpoints declaring none at all — the fact a reader most needs from this list. */
unguarded: number;
```

} | | inboundEdges | Record<string, number> | | outboundEdges | Record<string, number> | | flow | {
componentsWithFlow: number;

/**
* Components carrying at least one guard — NOT the number of guards.
*
* `ControlFlow.guards` is capped at six per component, so summing it produces an undercount
* presented as a total, which is the same sampling error `catchCounts` exists to avoid. How
* many components refuse work is exact whatever the cap is.
/
componentsThatRefuse: number;
catches: number;
byOutcome: Record<string, number>;
/* The member discarding the most failures silently. */
quietest?: { name: string; swallows: number };
}` |