What are the main differences between Aurelia and React, in terms of structure and maintainability?

Research question 1

## Introduction and Research Method

The purpose of this research is to compare Aurelia and React, focusing specifically on their structural approaches and long-term maintainability. Since the Sustainable Design team at Sorama currently uses Aurelia while most of the company has react as standard, understanding these differences is essential for making decisions about migration. The research method applied is desk research, analyzing official documentation, developer community insights, and technical comparisons from industry sources.

## Framework Analysis

### Aurelia

Aurelia is a convention-driven, MVVM-style front-end framework created by Rob Eisenberg. Its design philosophy emphasizes minimal boilerplate, strong data binding, and clean separation of concerns.

Structure:

* Uses two-way data binding a lot, reducing the need for explicit state updates.
* Encourages MVVM (Model–View–View Model) patterns, with clear separation between templates and logic.
* Provides convention-over-configuration: project structure is opinionated but flexible, relying on conventions rather than large configuration files.

Maintainability:

* Smaller ecosystem and less widespread community support compared to React.
* Long-term maintainability is harder in large organizations because fewer developers are trained in Aurelia.
* Good for small to medium projects with teams that value strong conventions.

### React

React is a UI library developed by Facebook (Meta), widely adopted in industry and supported by a large ecosystem of frameworks and tools (e.g., Next.js, Vite, Redux).

Structure:

* Emphasizes component-based architecture: everything is a composable component.
* Uses one-way data flow, with explicit state management via hooks or external libraries.
* Flexible and unopinionated: React only covers the view layer, leaving routing, state management, and styling to third-party solutions.

Maintainability:

* Very large developer community, extensive documentation, and long-term industry adoption.
* Rich ecosystem with many libraries and tools, increasing flexibility but requiring decisions on project setup.
* Easier to onboard new developers due to popularity and wide knowledge base.
* Supports long-term maintainability through predictable patterns (hooks, contexts) and large-scale best practices.

## Comparison Table

|  |  |  |
| --- | --- | --- |
| Aspect | Aurelia | React |
| Architecture | MVVM, convention-driven | Component-based, unopinionated |
| Data Binding | Two-way binding | One-way data flow |
| Ecosystem | Small, niche | Very large, mainstream |
| Flexibility | Moderate – conventions guide structure | High – developers choose tools and patterns |
| Maintainability | Harder long-term due to smaller community and limited library support | Easier long-term, strong ecosystem, easier onboarding |
| Learning Curve | Easier for simple apps due to conventions | Steeper at first (more decisions), but widely known |

## Conclusion

The main structural difference between Aurelia and React is that Aurelia enforces an MVVM, convention-driven approach with two-way binding, while React provides a flexible, component-based system with one-way data flow. In terms of maintainability, React clearly has the advantage: it benefits from a larger ecosystem, broader community support, and easier onboarding for new developers. Aurelia, while elegant in small projects, risks technical debt in larger organizations due to its niche adoption and limited library support.  
For Sorama, migrating from Aurelia to React not only aligns with company-wide practices but also ensures better long-term maintainability, collaboration, and scalability.