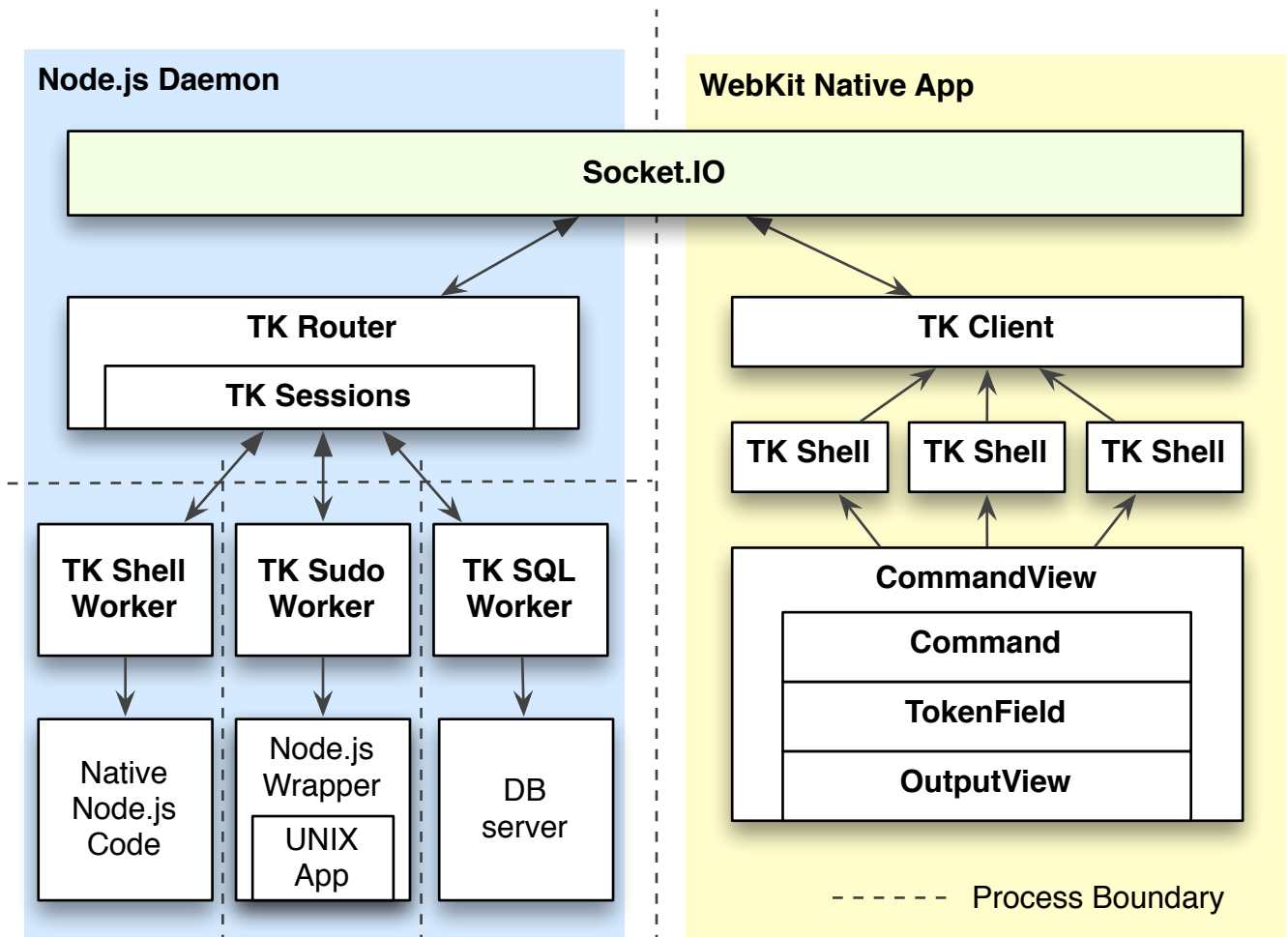


TermKit Architecture

TermKit is a reinvention of the Unix command-line. Its goals are:

- 1) Ditch plain-text input. Use a token-based system with instant feedback and inline completion.
- 2) Evolve output beyond the monospace character grid; use rich media and design.
- 3) Separate the interactive output for humans from structured data being piped around.
- 4) Run the UI locally to avoid network lag. Allow for asynchronous multiplexing of commands.
- 5) Use sane defaults for commands, make Unix more discoverable.



The **node.js** process runs **the back-end**, doing the actual work. It launches new commands and processes on-demand, and sends the results to the front-end .

Multiple session types can be meshed together, e.g. a shell console next to an SQL console.

Every system user runs their own instance of the node.js process. TermKit does not replace native authentication.

The **WebKit-based front-end** lets users enter commands and interact with the resulting output

Command input revolves around tokenized input with real-time feedback.

Command output consists of a basic layout framework and a widget set. This ranges from text display to rich media to dense infographics.