

Every HTTP stack
is an intermediary

HTTP Workshop 2024
Kazuho Oku

A commonly heard story

- “my HTTP client does not support informational responses”

A commonly heard story

- “my HTTP client does not support informational responses”
- “my HTTP client does not forward response incrementally to my application”

A commonly heard story

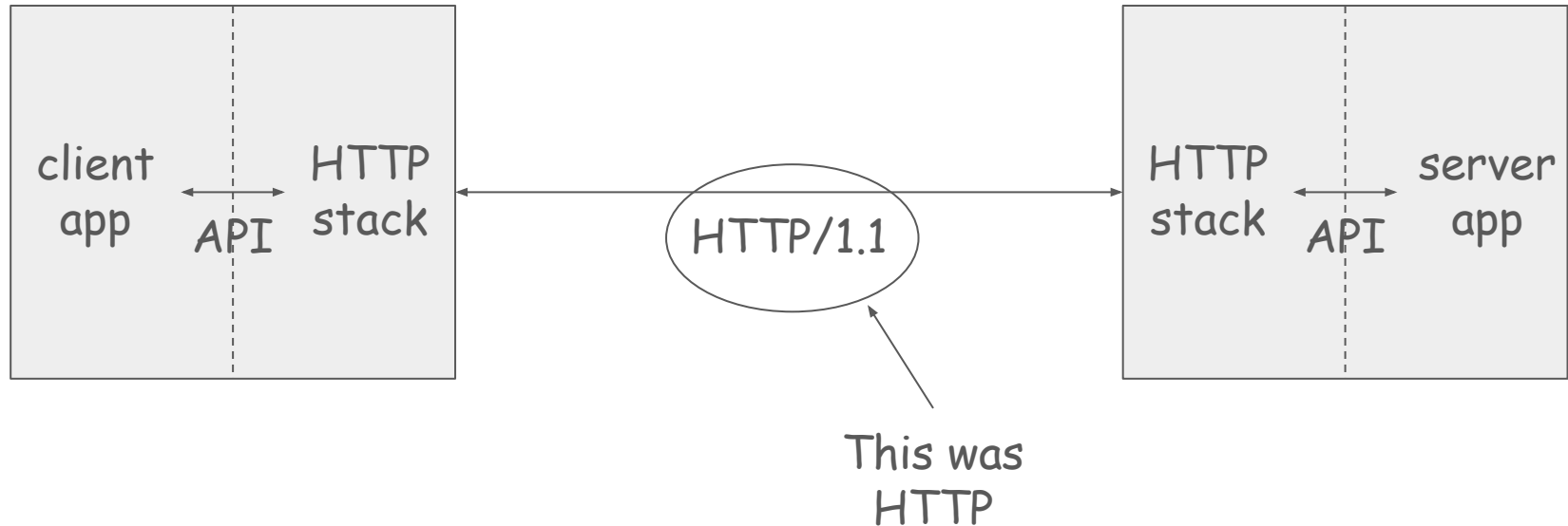
- “my HTTP client does not support informational responses”
- “my HTTP client does not forward response incrementally to my application”

What is the assumption behind these complaints?

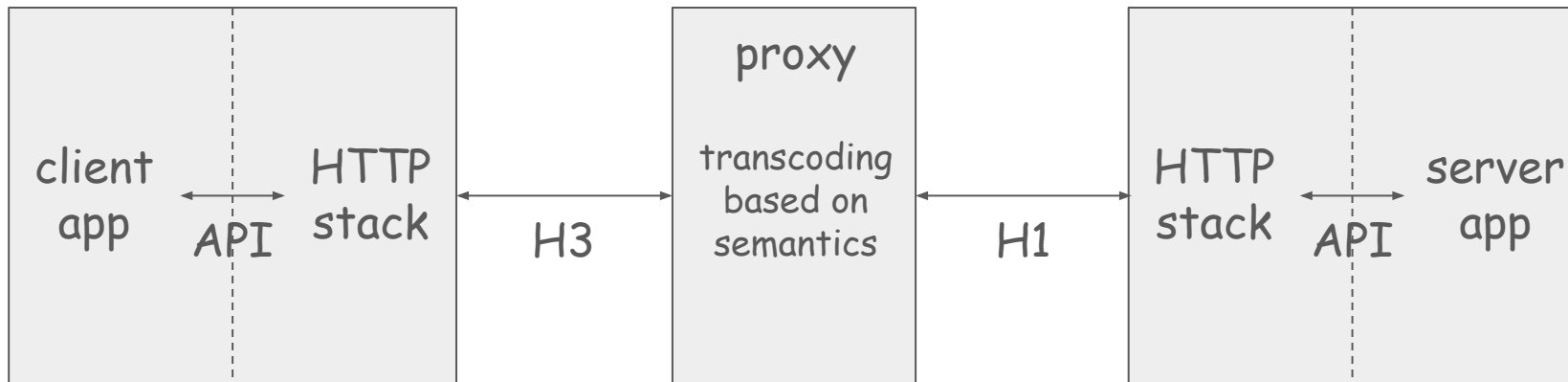
HTTP started as wire protocol



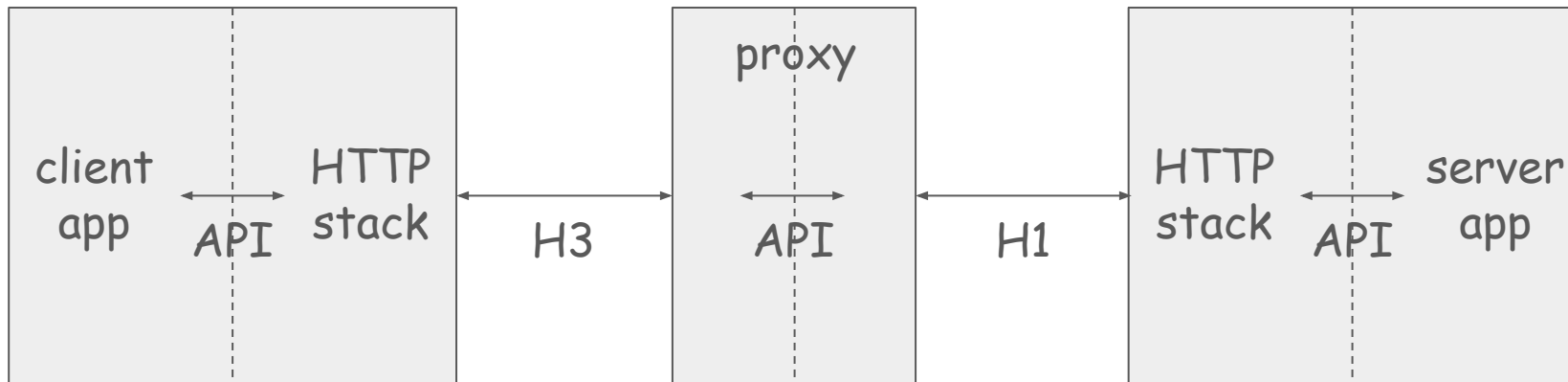
HTTP started as wire protocol



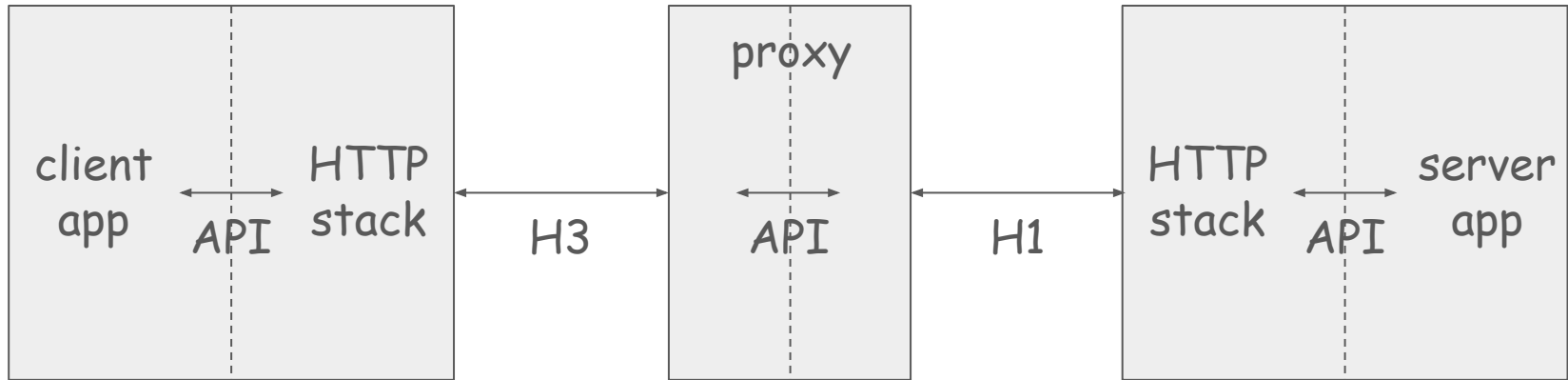
split of semantics and wire format



split of semantics and wire format



semantics is the API



semantics is the requirements for low-level API

- to fulfill the needs of various applications built on top of HTTP, HTTP stacks need to provide access to all the capabilities promised by the HTTP semantics

semantics is the requirements for low-level API

- to fulfill the needs of various applications built on top of HTTP, HTTP stacks need to provide access to all the capabilities promised by the HTTP semantics
- HTTP stacks can provide extra transformations or restrictions:
 - add cookies
 - policy enforcement (e.g., origin)
 - compression

semantics is the requirements for low-level API

- to fulfill the needs of various applications built on top of HTTP, HTTP stacks need to provide access to all the capabilities promised by the HTTP semantics
- HTTP stacks can provide extra transformations or restrictions:
 - add cookies
 - policy enforcement (e.g., origin)
 - compression

semantics is the requirements for low-level API

- makes sense?
- should we be more vocal about it?