Web architecture

Distributed systems and services

What is architecture?

- the relationship of things
- In IT
  - Data
  - Procedures

Component terms

- Client
  - any piece of software that requests something
  - any piece of hardware that is running client software
- Server
  - any piece of software that responds to a request
  - any piece of hardware that is running server software
- Host
  - the server on the internet

Client - browser operations

- make http requests
- bundle up form data
- execute JavaScript
  - process user interactions, update DOM
- parse and render html & CSS
- allow plugins to extend features
- cache/retrieve local files
- render XML, RSS, SVG, GZIP…
- launch apps for other files

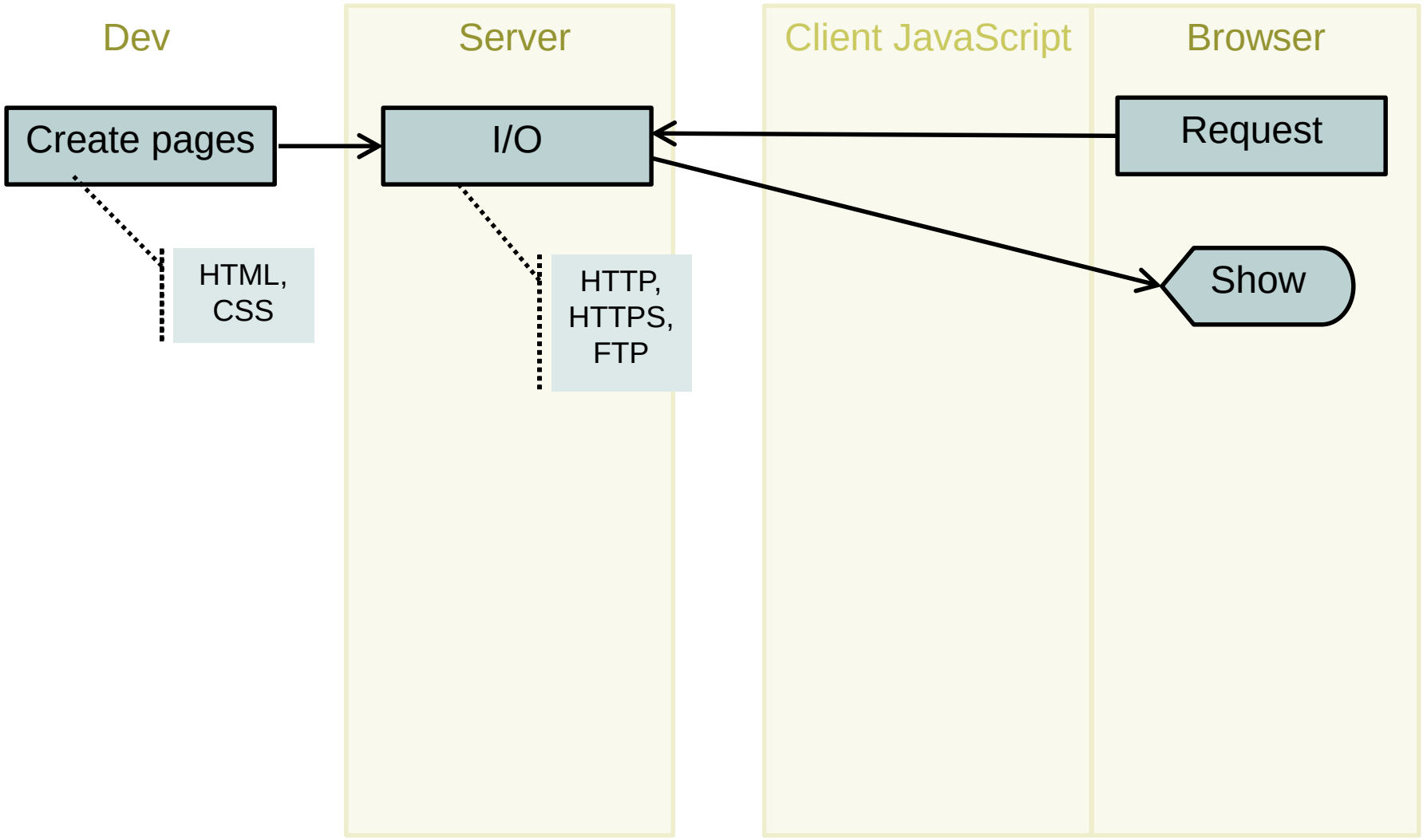
Web server types

- Asynchronous – best choice
  - Nginx (production)
  - node.js with Express (dev & testing)
- Apache HTTPD
  - original free web server
- Microsoft based
  - IIS, runs php
- Java based
  - Tomcat, any application server e.g. Weblogic

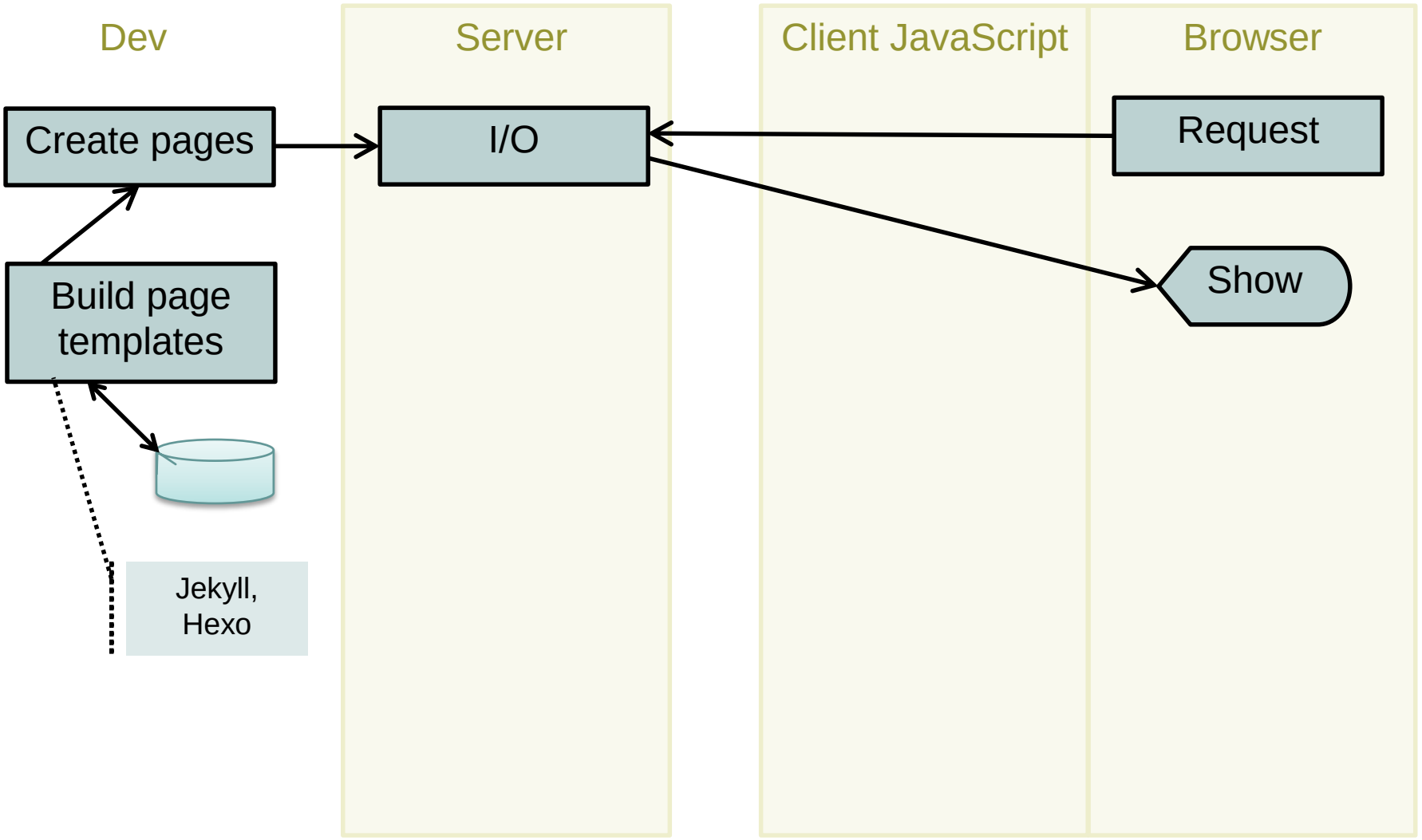
Server operations - web

- listen/respond to http request
- create a header
- get and send files
- handle email, ftp
- execute a language –build web pages
  - process routing
  - process page templates
  - process authentication rules
  - talk to DBMS
- manage multiple web sites
- store session data

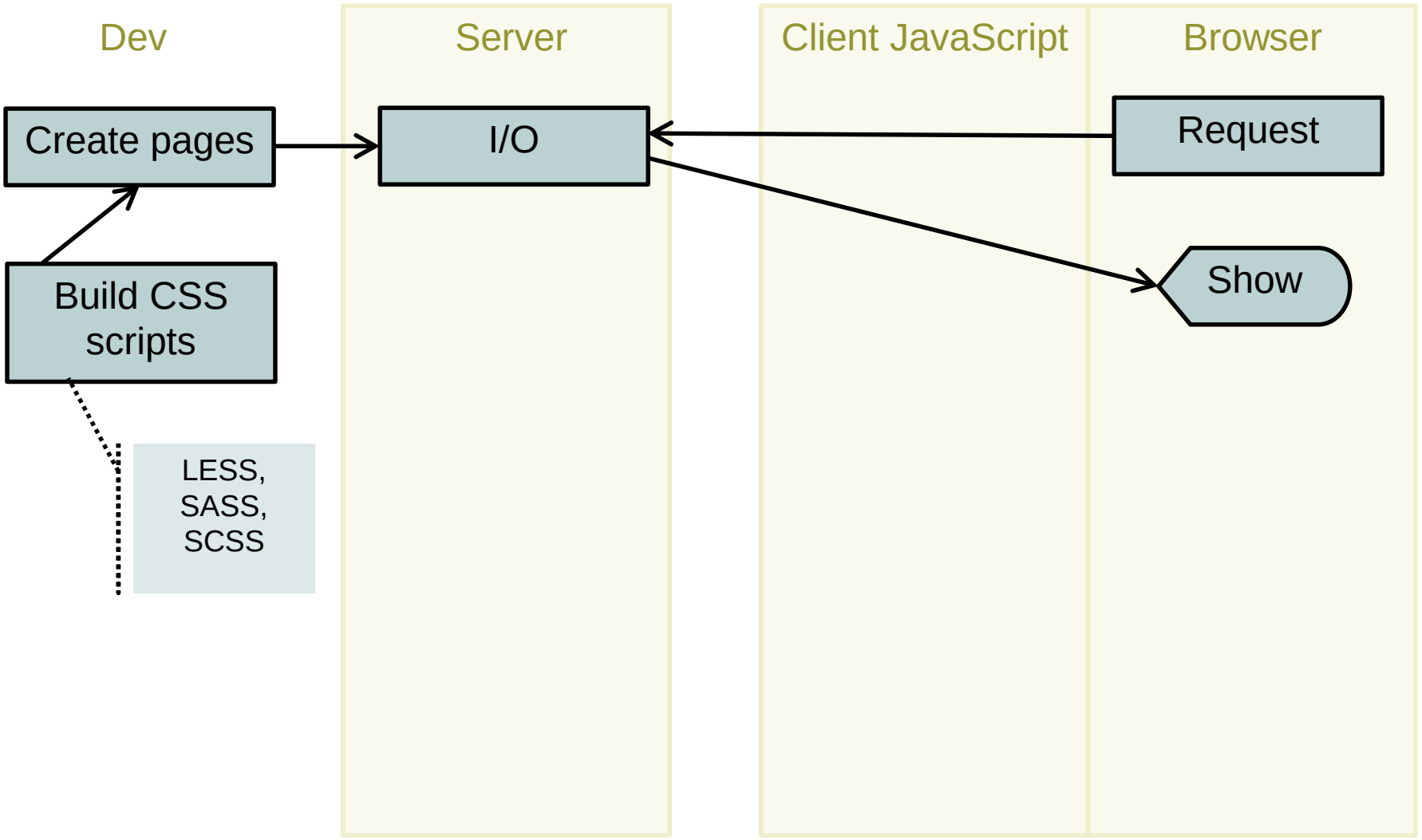
Web 1.0


Dev
Server
Client JavaScript
Browser
Create pages
I/O
Request
Show
HTML,
CSS
HTTP,
HTTPS,
FTP

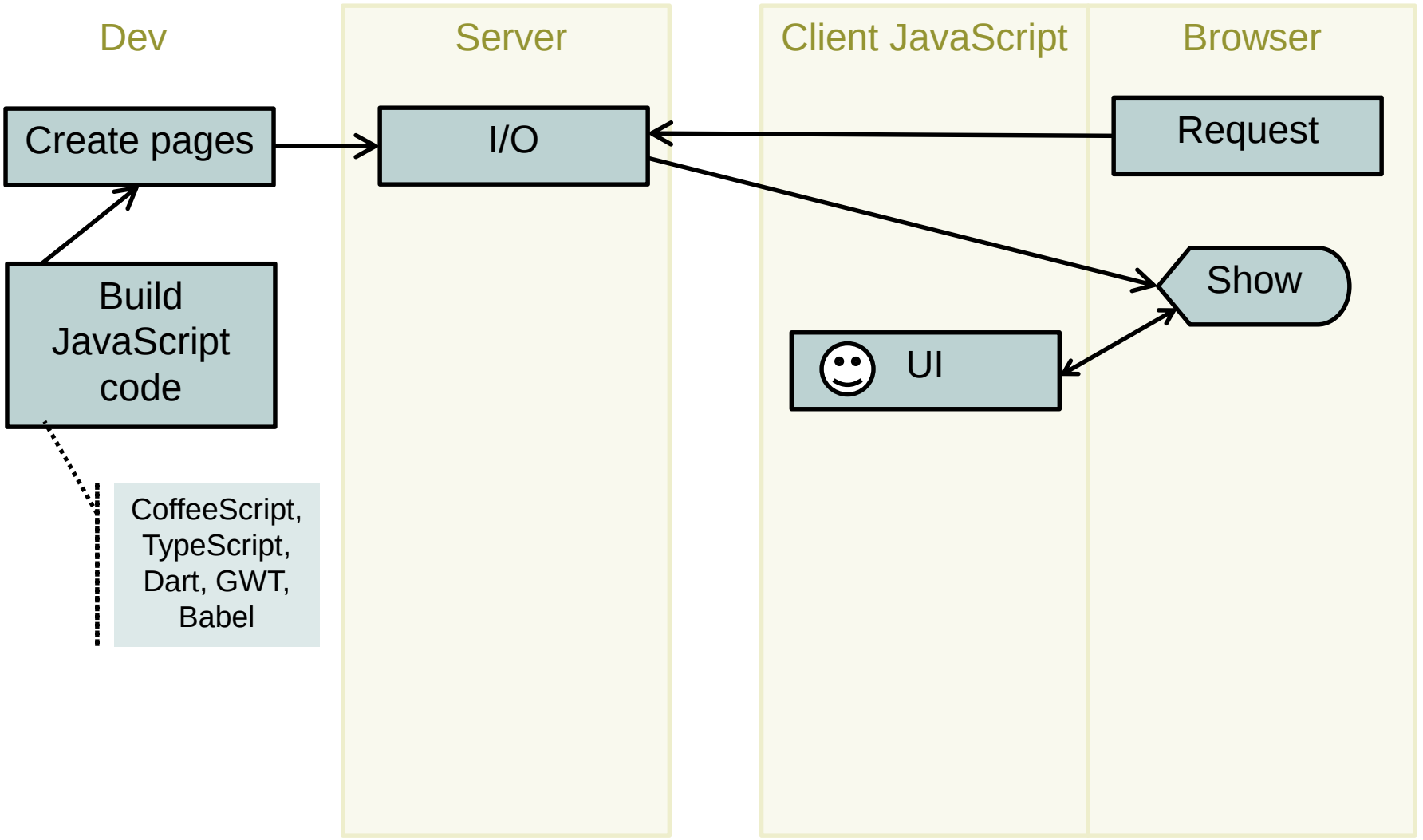
Web 1+ – static site generators



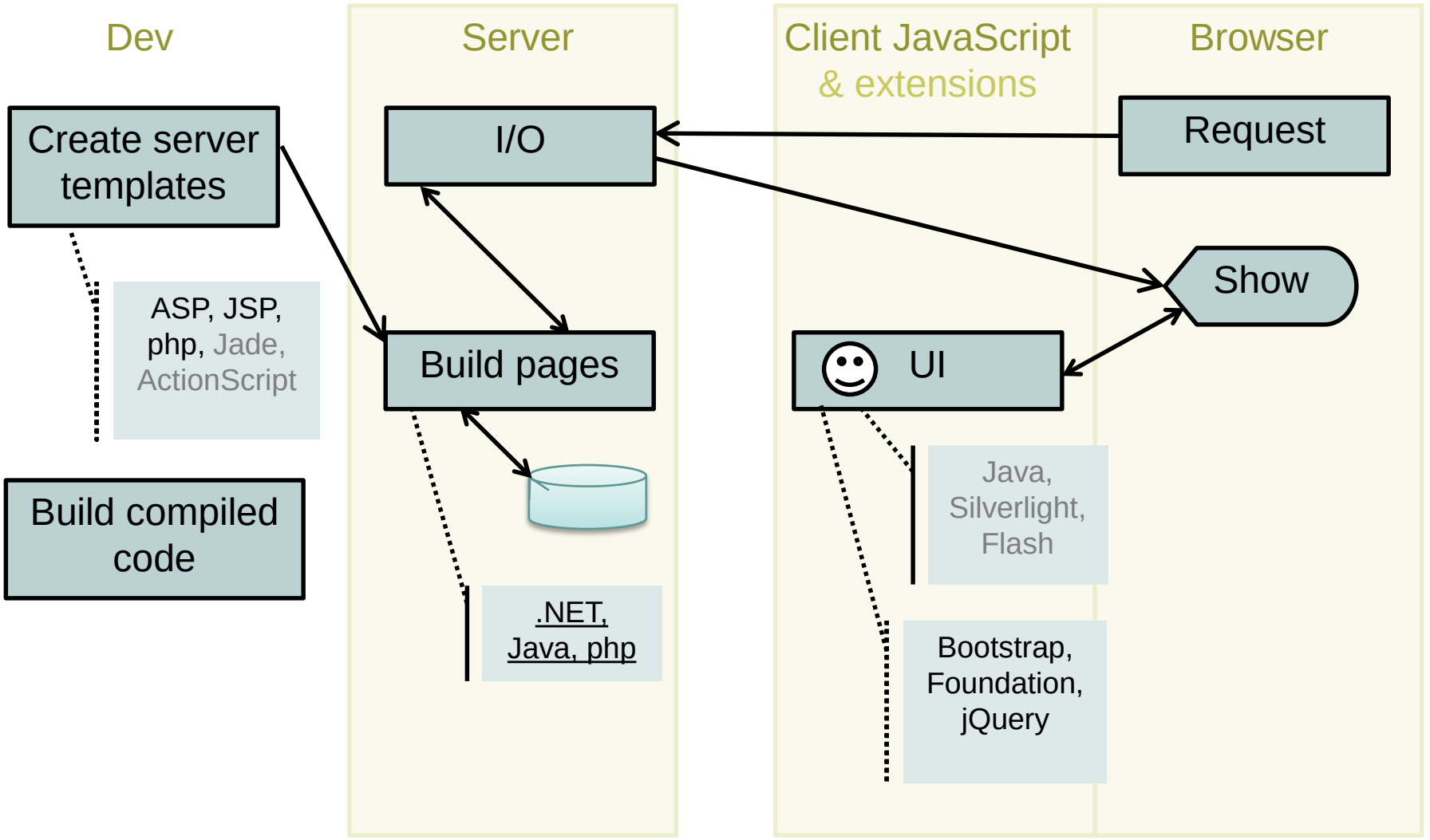
Web 1+ – CSS generators



Web 1+ – JavaScript generators



Web 2.0 – dynamic web sites



Pros / Cons

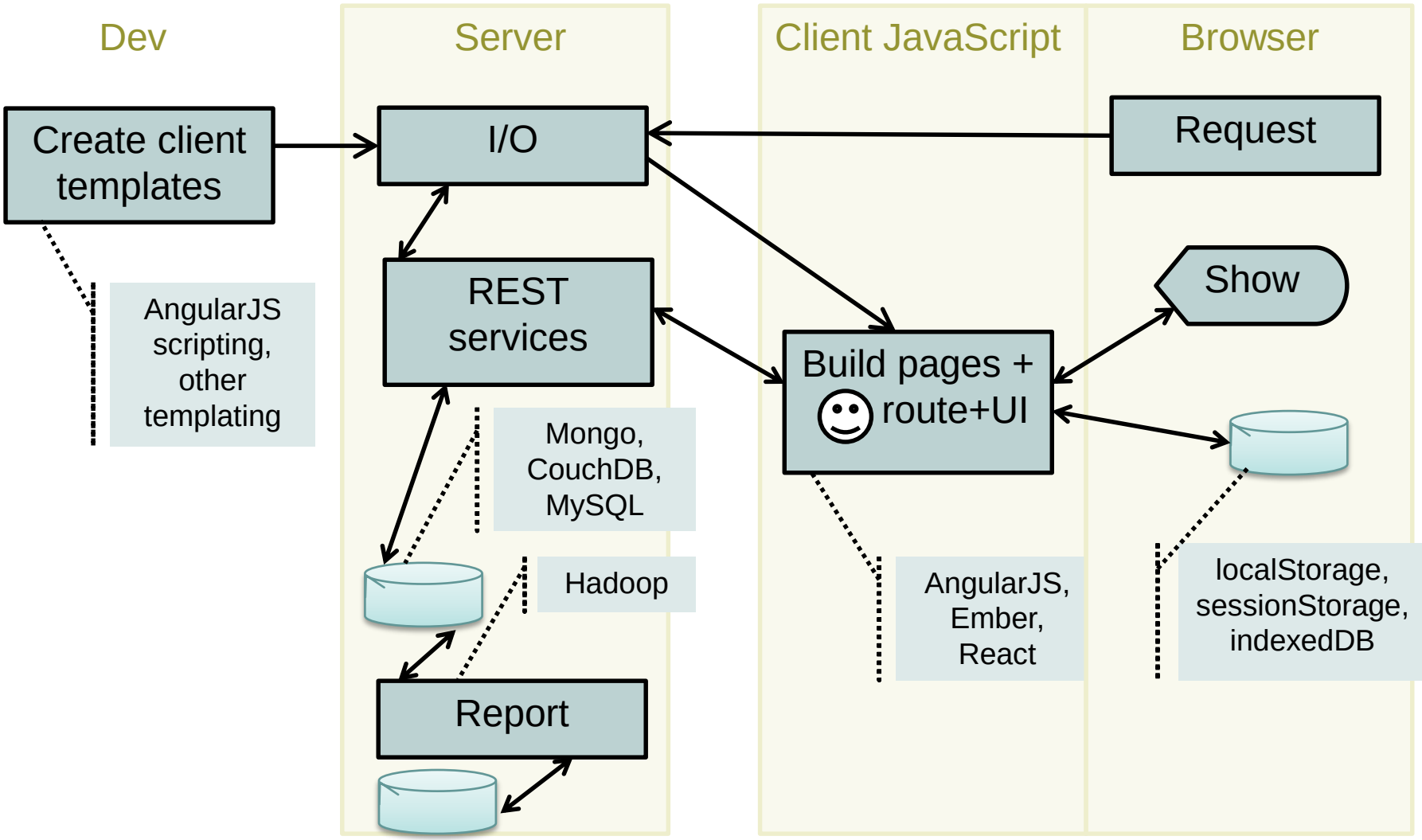
Application

- **Pro** – Simple at first, single codebase, resource efficient when small
- **Con** – team growth creates coordination overhead, poor enforcement of modularity, poor vertical scaling, all-or-nothing deploy, long build times

Database

- **Pro** – simple at first, join queries easy, transactions, resource efficient when small
- **Con** – coupling over time, performance & scalability bottleneck, difficult to tune, single point of failure

Web 3 – apps


Dev
Server
Client JavaScript
Browser
Create client templates
I/O
Request
AngularJS scripting, other templating
REST services
Show
Build pages + route+UI
Mongo, CouchDB, MySQL
Hadoop
AngularJS, Ember, React
localStorage, sessionStorage, indexedDB
Report

Designing a microservice

- “Microservices are nothing more than SOA done properly.” - Randy Shoup from YOW15
- A service (application)
  - one purpose
  - self-contained
  - independent
  - clearly defined interface
  - isolated persistence (even to the point of having a database per service)

SPA overview

SPA

- Pro – simple units, independent scaling, performance, testing & deployment, tuning
- Con – many units, latency, no transactions, refactoring, requires better tooling

Prerequisites

- process maturity – rapid release, automated testing, CI
- organizational maturity – design like org, modular system means small teams, org needs refactoring
- operational maturity – team to service, cross-functional, end-to-end ownership (DevOps “You build it, you run it”), monitoring

Web server communication protocols

- Web 1
  - http, https, ftp
- Web 2
  - XHR (AJAX)
- Web 3
  - WebSockets (.NET's SignalR prefers)
  - Server push (server-sent events SSE) – no IE/Edge
  - WebRTC peer-to-peer
    - https://webrtc.org/
    - Also stream video, access photo, audio

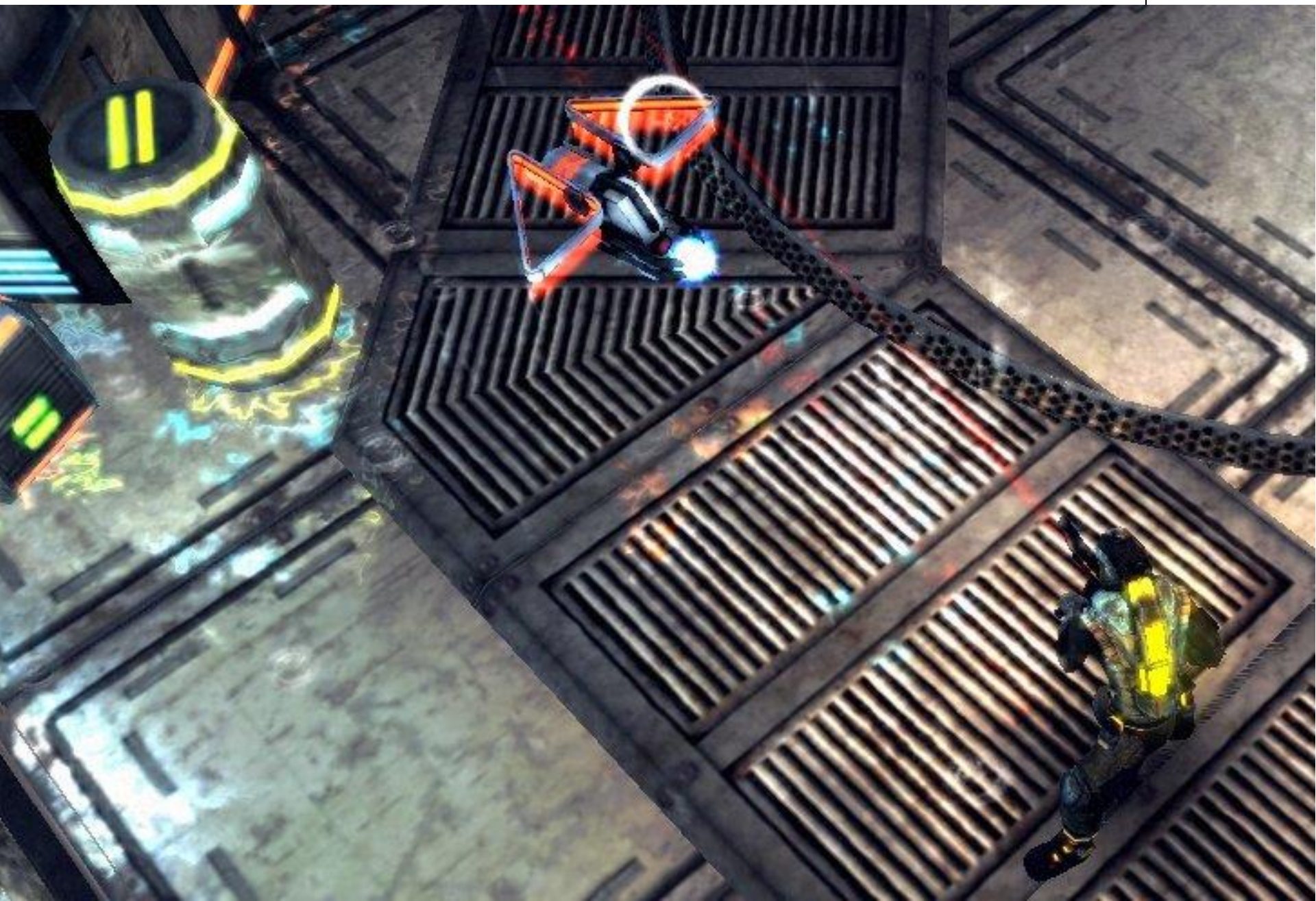
Web 4 – compiled apps

- the final maturity stage of an execution environment
- W3C WebAssembly
  - http://webassembly.org/
  - Browser Preview – Oct 2016
  - Final version in all browsers 1Q 2017
  - Compiled code up to 20x faster
  - View Source support
  - C++, JavaScript, Java, etc.
  - API for device support

Web 3/4+ serverless

- a distributed application strategy for speed
- Serverless
  - microservices run independently as needed
  - AWS Lambda - https://aws.amazon.com/lambda/
  - Azure Functions - https://azure.microsoft.com/en-us/services/functions/
  - Google Cloud Functions - https://cloud.google.com/functions/
    - written in JavaScript, execute in Node's V8

http://webassembly.org/demo/



Resources

- Randy Shoup – Pragmatic Microservices
  - http://yowconference.com.au/slides/yow2015/Shoup-PragmaticMicroservices.pdf
- Building Microservices by Sam Newman
  - Feb 2015