

# Web Development in D

Vladimir Panteleev

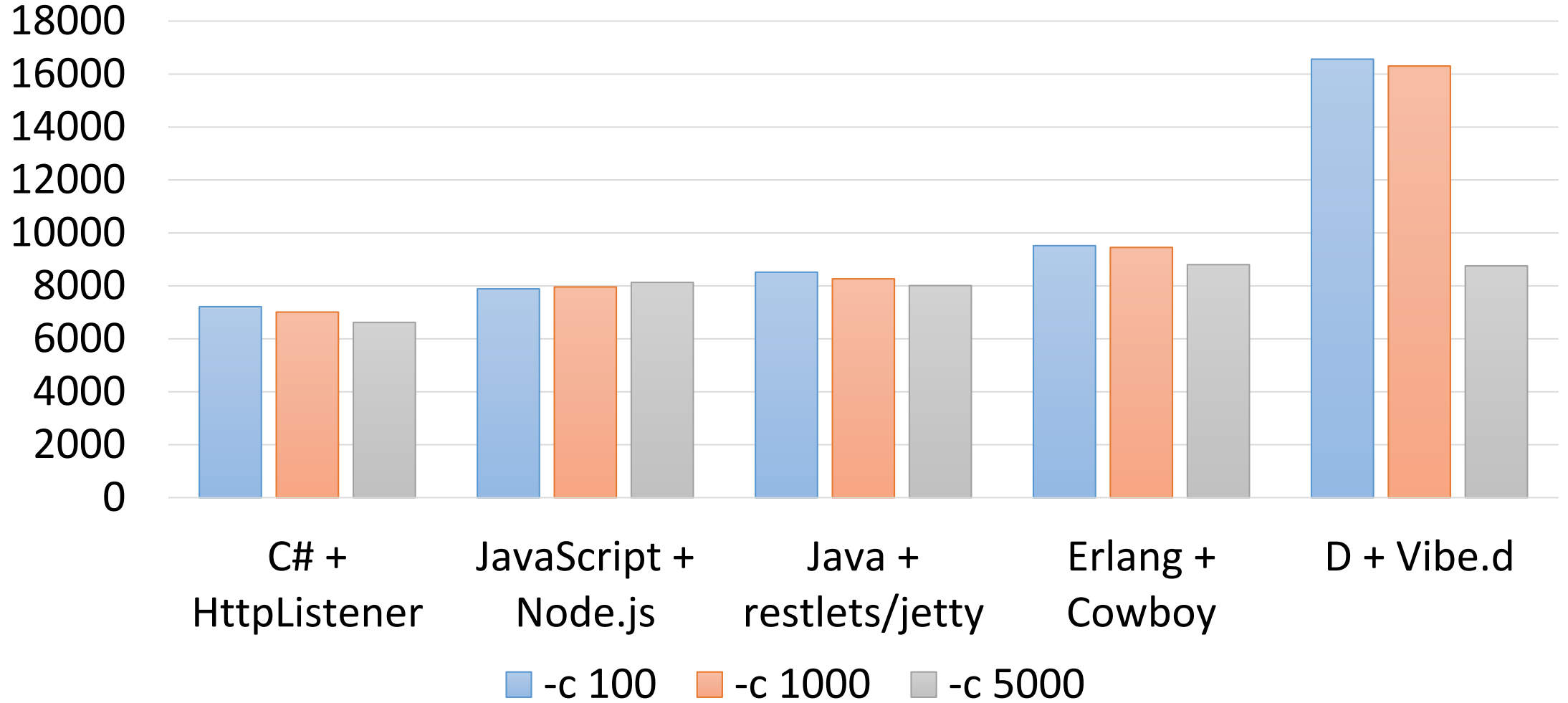
DConf 2013

# What is web development?

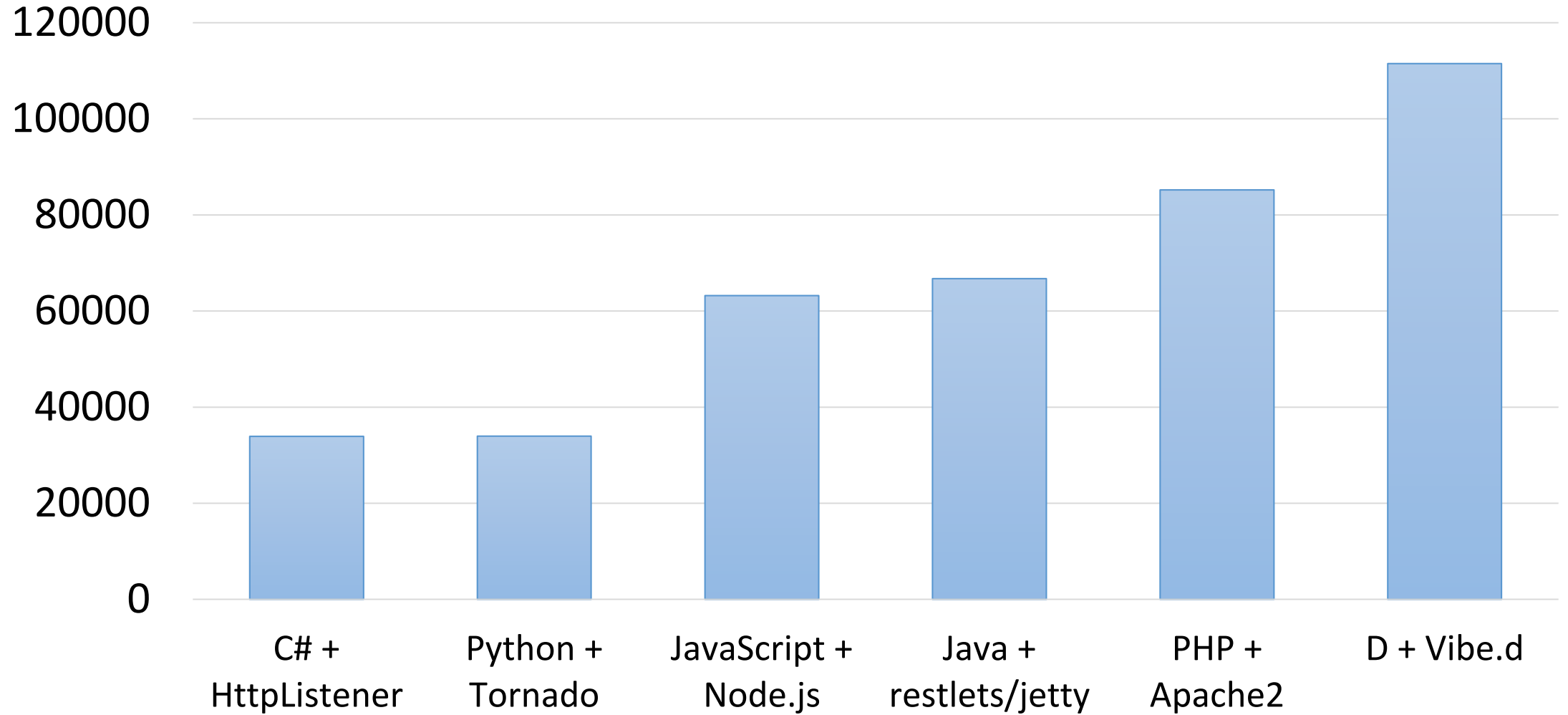
- Websites
- Web applications
- Web services (REST)
- Databases

Why D?

# Requests per second



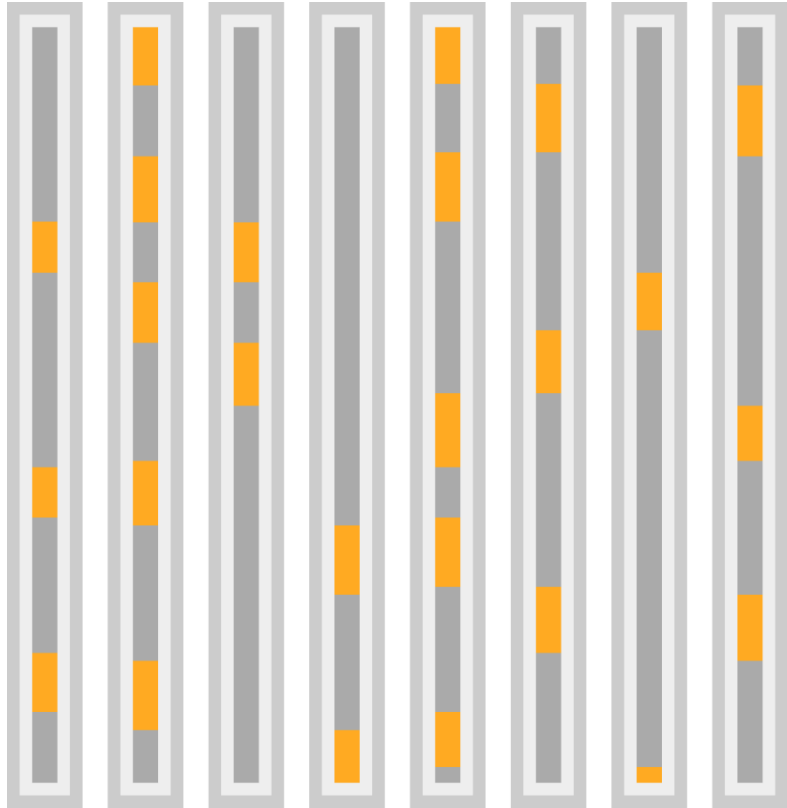
# Requests per second



# vibe.d

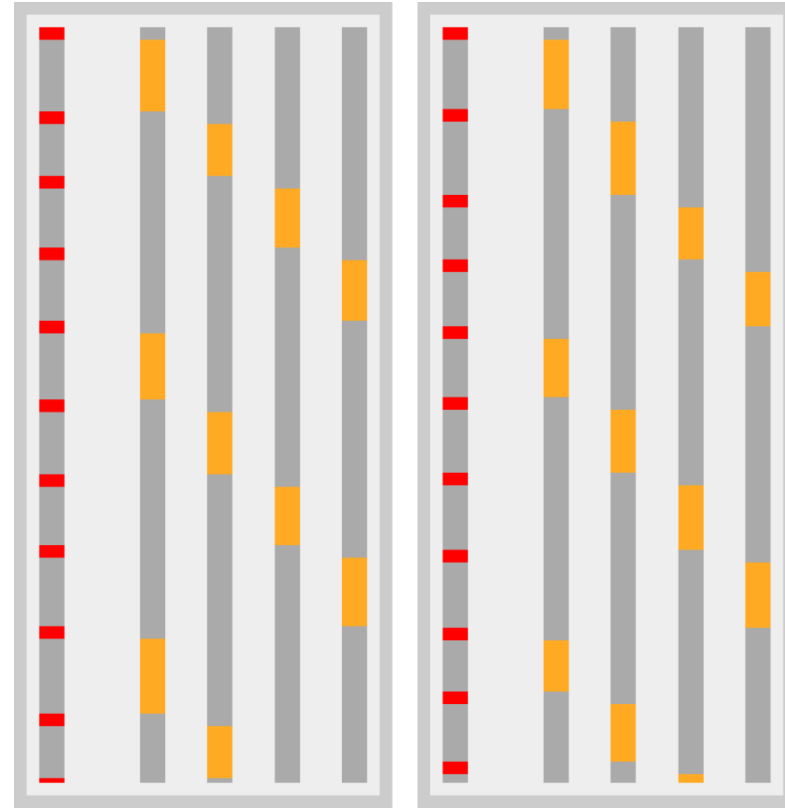
- Asynchronous I/O, synchronous API
- Exception-based error handling
- Compile-time templates
- Load-balancing (WIP)
- Zero-downtime changes (WIP)
- REST, SSL, MongoDB, Redis

# Synchronous



One thread (or process) per connection

# Asynchronous



Multiple connections per thread (or process)

# Synchronous

```
// ...  
auto result = fun();  
// use result
```

# Asynchronous

```
// ...  
fun(delegate(result) {  
    // use result  
});
```



# Synchronous

```
logInfo("Connecting to DB...");
auto db = connectMongoDB("localhost")
    .getDatabase("test");
auto coll = db["test"];

logInfo("Inserting value...");
coll.insert(["a" : 2]);

logInfo("Querying DB...");
foreach (doc; coll.find(["name" : "hans"]))
    logInfo(doc.toJson().toString());
```

# Asynchronous

```
console.log('Connecting to DB...');
client.open(function(err, p_client) {
    client.collection('test', function (err, coll) {
        if (err) throw err;

        console.log('Inserting value...');
        coll.insert({a:2}, function(err, docs) {
            if (err) throw err;

            console.log('Querying DB...');
            var q = coll.find({"name" : "hans"});
            q.toArray(function(err, results) {
                if (err) throw err;

                for (var i=0; i<results.length; i++)
                    console.log(JSON.stringify(results[i]));
            });
        });
    });
});
```

# vibe.d – Intro

```
$ dub init appname  
$ cd appname  
$ dub
```

```
appname/  
  package.json  
  source/  
    app.d  
  public/  
    images/  
    styles/  
      style.css  
  views/  
    layout.dt  
    index.dt
```

## — package.json —

```
{  
  "name": "appname",  
  "description": "My fabulous new app",  
  "copyright": "Copyright (C) ...",  
  "homepage": "http://appname.org",  
  "authors": [  
    "Hans Wurst"  
  ],  
  "dependencies": {  
    "vibe-d": ">=0.7.12"  
  }  
}
```

## — app.d —

```
import vibe.d;

void index(HTTPRequest req,
           HTTPServerResponse res)
{
    res.renderCompat!("index.dt",
                    HTTPServerRequest, "req")(req);
}

shared static this()
{
    auto router = new URLRouter;
    router.get("/", &index);

    auto settings = new HTTPServerSettings;
    settings.port = 8080;
    listenHTTP(settings, router);
}
```

## — index.dt —

```
extends layout

block body
    h1 Example page - Home
    p Hello, World!
```

## — layout.dt —

```
!!! 5
html
    head
        title Example page
    body
        block body
```

# Compile-time “Diet” templates

```
!!! 5
html
  head
    title My page: #{pageTitle}
  body
    h1= pageTitle
    p This is the content of this page. The
      | title "#{pageTitle}" is inserted dynamically.
      | We can also use loops and other D statements:
    block placeholder
    p
      - foreach(i, ch; pageTitle)
        | #{i+1}. character: #{ch}
    p(style='color: red;') This text is red.
    p.special.small This paragraph has the 'special'
      | and 'small' CSS classes
    p#footer This paragraph has the id 'footer'.
    #somediv
      | I'm a text inside the div ##somediv
```

# REST (Representational State Transfer)

- Client-server
- Stateless
- Cacheable
- Layered system
- Uniform interface

HTTP request examples:

```
GET /items/
```

```
GET /items/42
```

```
PUT /items/42
```

```
DELETE /items/42
```

# REST client

```
interface IExampleAPI
{
    // Matches "GET /"
    string getIndex();

    // Matches "GET /data"
    @property string data();

    // Matches "PUT /data"
    @property void data(string info);

    // Matches "POST /sum"
    // or "GET /sum?a=...&b=..."
    int postSum(int a, int b);

    // Matches "GET /item/<category>/<item>"
    @path("item/:category/:item")
    int getItem(string _category, int _item);
}
```

```
auto api = new RestInterfaceClient!
    IExampleAPI("http://localhost/");
```

```
auto index = api.getIndex();
api.data = "My data";
assert(api.data == "My data");
assert(api.postSum(2, 3) == 5);
```

# REST server

```
class Example : IExampleAPI
{
  override:
    string getIndex() { return "Index!"; }

    string _data;
    @property string data() { return _data; }
    @property void data(string v) { _data=v; }

    int postSum(int a, int b) {
      return a + b;
    }

    int getItem(string _category,
               string _item) {
      // ...
    }
}

auto routes = new URLRouter;
registerRestInterface!IExampleAPI(
  routes, new Example(), "/"
);
```



- SSL: via OpenSSL
- Database support: MongoDB, Redis
- Root (setuid)
- dub registry:
  - socket.io (JS real-time communication library)
  - sass-vibe (CSS extension)
  - vibedist (load balancing, virtual hosts, zero-downtime changes)
  - mysql-native (native MySQL driver for vibe.d)
  - zeal (MVC framework on top of vibe.d)
  - and more: <http://registry.vibed.org/>

<http://vibed.org/>

# HibernateD ORM

```
class Customer {
    int id;
    string name;

    // @Embedded is inferred -
    // Address is @Embeddable
    Address address;

    // @ManyToOne inferred
    Lazy!AccountType accountType;

    // @OneToMany inferred
    User[] users;

    this() {
        address = new Address();
    }
}
```

```
class User {
    long id;
    string name;
    Customer customer;

    @ManyToMany
    LazyCollection!Role roles;
}

class Role {
    int id;
    string name;

    @ManyToMany
    LazyCollection!User users;
}
```

```
@Embeddable
class Address {
    string zip;
    string city;
    string streetAddress;
}

class AccountType {
    int id;
    string name;
}
```

<http://sourceforge.net/p/hibernated/>

# HibernateD ORM

```
// create sample data
```

```
Role r10 = new Role();
r10.name = "role10";
Role r11 = new Role();
r11.name = "role11";
Customer c10 = new Customer();
c10.name = "Customer 10";
User u10 = new User();
u10.name = "Alex";
u10.customer = c10;
u10.roles = [r10, r11];
sess.save(r10);
sess.save(r11);
sess.save(c10);
sess.save(u10);
```

```
// load and check data
```

```
User u11 = sess
    .createQuery("FROM User WHERE name=:Name")
    .setParameter("Name", "Alex")
    .uniqueResult!User();

assert(u11.roles.length == 2);
assert(u11.roles[0].name.startsWith("role"));
assert(u11.roles[1].name.startsWith("role"));
assert(u11.customer.name == "Customer 10");
assert(u11.customer.users.length == 1);
assert(u11.customer.users[0] == u10);
assert(u11.roles[0].users.length == 1);
assert(u11.roles[0].users[0] == u10);
```

<http://sourceforge.net/p/hibernated/>

# Other projects

- Web development:
  - Adam Ruppe's arsd – <http://bit.ly/12sul2N>
  - DFeed / ae – <https://github.com/CyberShadow/ae>
- Package management:
  - Orbit – <https://github.com/jacob-carlborg/orbit>
- Database access:
  - ddbc – <http://dsource.org/projects/ddbc>
- Template engine:
  - Mustache for D – <http://dsource.org/projects/ddbc>

# D and JavaScript

- JavaScript in D

- DMDScript - <http://www.digitalmars.com/dscript/>
- Higgs

- D to JavaScript

- Adam Ruppe's DMD hack: <http://bit.ly/zlZOMl>
- ABA's LDC+Emscripten+Asm.js experiments: <http://j.mp/YUCtDv>

- Haxe - <http://www.dsource.org/projects/haxed/>

# Future

More Vibe!

<https://github.com/rejectedsoftware/vibe.d>

<http://registry.vibed.org/>

core.event / std.event ?

[http://wiki.dlang.org/Event\\_system](http://wiki.dlang.org/Event_system)

# Thank you

Acknowledgements:

Sönke Ludwig – Vibe.d and feedback  
Михаил Страшун (Dicebot) – benchmarks

# Questions?