



You are an engineer at `Acme Health` and your team is building a new Dashboard product...

◆ JIRA-9845

As a user of Acme Health

I want a personalized dashboard\*

So that I can track my health metrics

*\*Mockups Attached*



The team is ready to get started... but which  
frontend framework will you choose?



# Why Vue.js?

- Use modern / future JavaScript today
- Great developer experience
- Best tooling available ([@vue/cli](#))
- Easy code splitting support
- Benefits our clients (*speed, efficiency, etc.*)
- Extensive community support – #1 in fact!

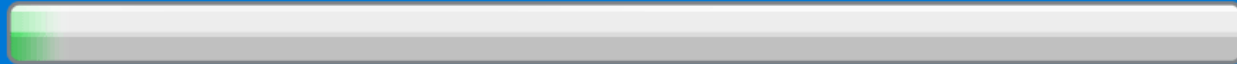


...so what's stopping us?





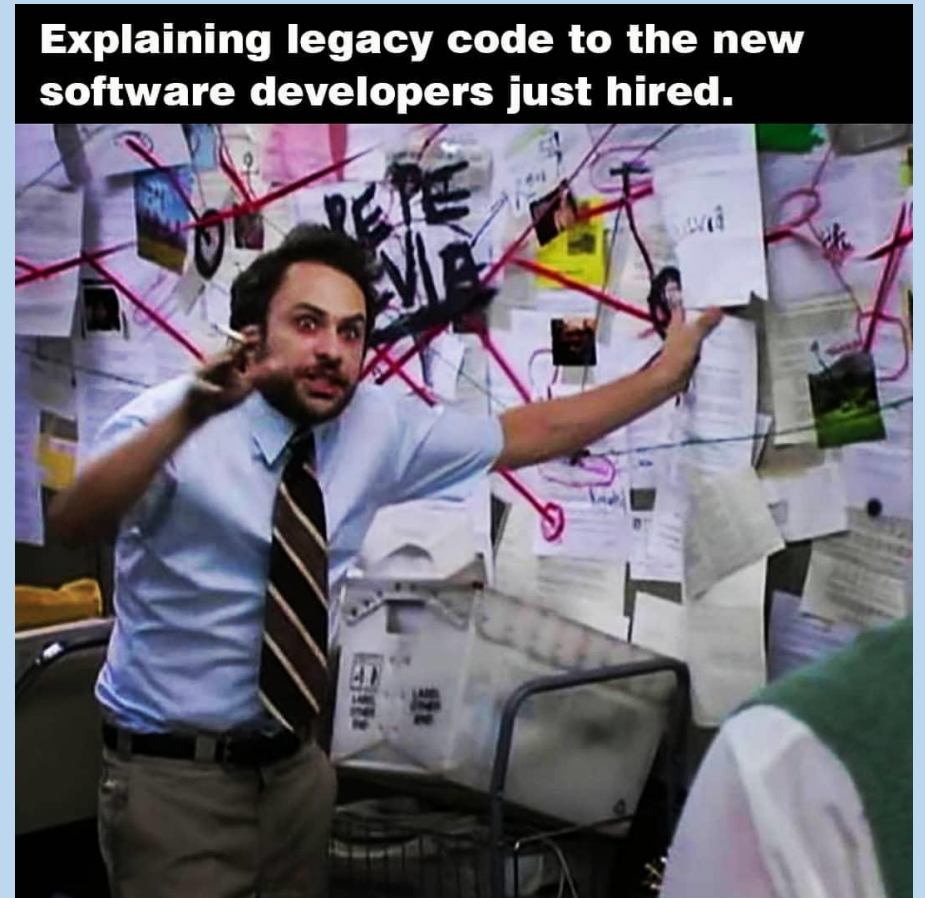
Internet Explorer has stopped working





# Existing codebases can hinder adoption

- Existing or legacy applications
- MVC, Web Forms, PHP...
- Already using a SPA framework
- Lack of `npm` build process
- Browser support requirements
- Not always a clear path forward



# So, what are our options?

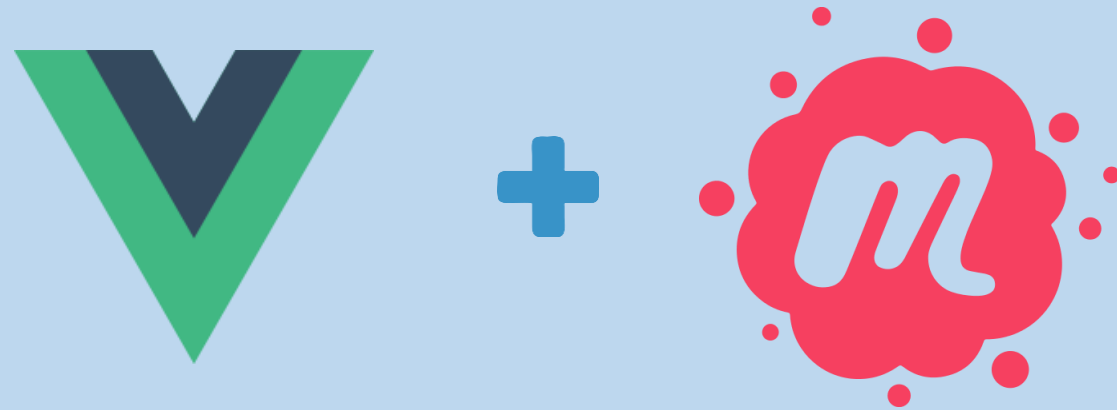
rewrite everything



build a separate app



# Incrementally Adopting Vue.js



presented by...



**Brandon Smith**  
Director of Software Engineering



**Ruslan Popovych**  
Senior Software Architect





# Monolithic Ext JS App



# Our approach to `Incremental Adoption`

- Inspired by [micro-frontends.org](https://micro-frontends.org)
- Enabled by strong CI / CD capabilities
- Motivated by desire to use modern tooling
- Provides developer efficiencies
- Benefits our clients

```
// setup
```

```
$ vue create meetup  
$ cd meetup  
$ npm run build
```

```
// main.js
```

```
import Vue from 'vue'  
import App from './App.vue'
```

```
new Vue({  
  render: h => h(App),  
}).$mount('#app')
```

```
// index.html
```

```
<!DOCTYPE html>  
<html lang=en>  
  
<head>  
  <meta charset=utf-8>  
  <link rel=icon href=/favicon.ico> <title>tmp</title>  
  <link href=/css/app.e2713bb0.css rel=preload as=style>  
  <link href=/js/app.f594fb06.js rel=preload as=script>  
  <link href=/css/app.e2713bb0.css rel=stylesheet>  
</head>  
  
<body>  
  <noscript><strong>Redacted...</strong></noscript>  
  <div id=app></div>  
  <script src=/js/chunk-vendors.ecd76ec1.js></script>  
  <script src=/js/app.f594fb06.js></script>  
</body>  
</html>
```



# Integration Challenges

1. How do we expose our Vue.js apps for consumption?
2. How can we maintain runtime advantages:
  - Code splitting & dynamic loading
  - Aggressive cache control

# Basic Principle

- All you need is a **div**
- The **div** can be anywhere
- Data can be passed in
- Listeners can be passed in

```
<div id="mount">  
  <!-- App Mount Point -->  
</div>
```

```
new Vue({  
  template: '<p> Hello, {{ name }} </p>',  
  el: '#mount',  
  data: {  
    name: 'Meetup'  
  }  
})
```

# Mount Functions

- Remove `index.html` - rendering inside existing app(s)
- Provide means to mount to ANY `div` at runtime
- No automatic bootstrapping of application
- Flexibility to provide data, props, etc

```
// main.js
```

```
window.mount.helloMeetup = (el, config) => {  
  return new Vue({  
    el,  
    render: h => h(() => import('@components/Hello', config))  
  })  
}
```

```
// legacy-app.html
```

```
<script src=/main.js></script>  
<div id="mount">  
  <!-- App Mount Point -->  
</div>
```

```
<script>  
  window.mount.helloMeetup(document.querySelector('#mount'))  
</script>
```

# Application Manifests

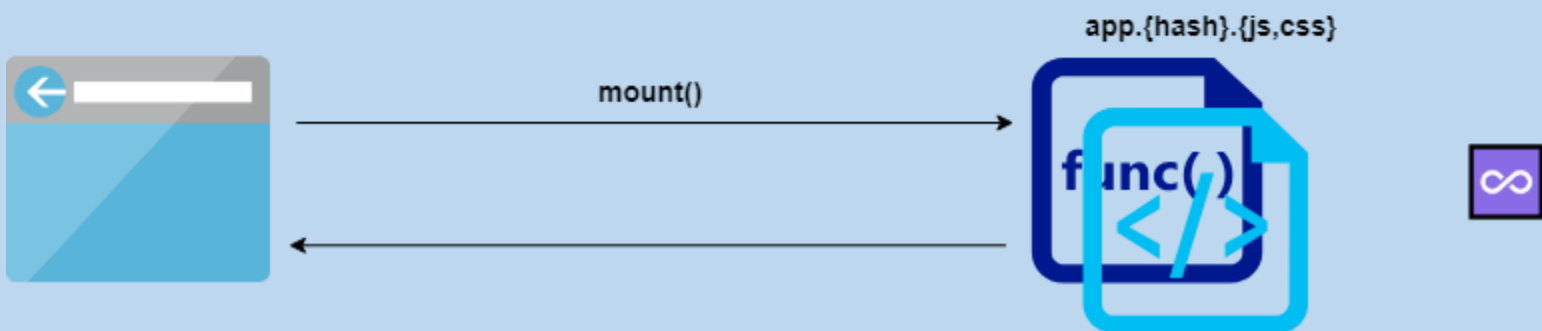
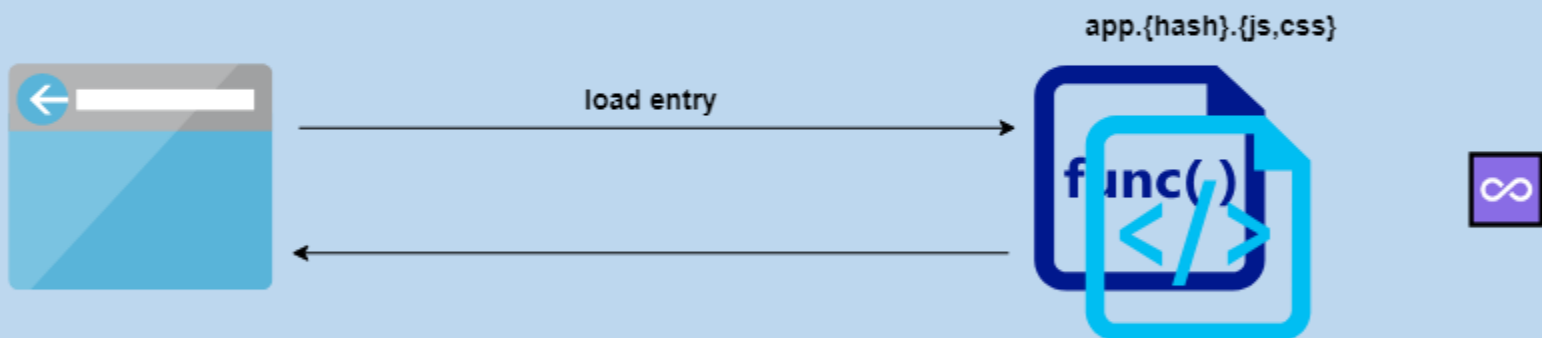
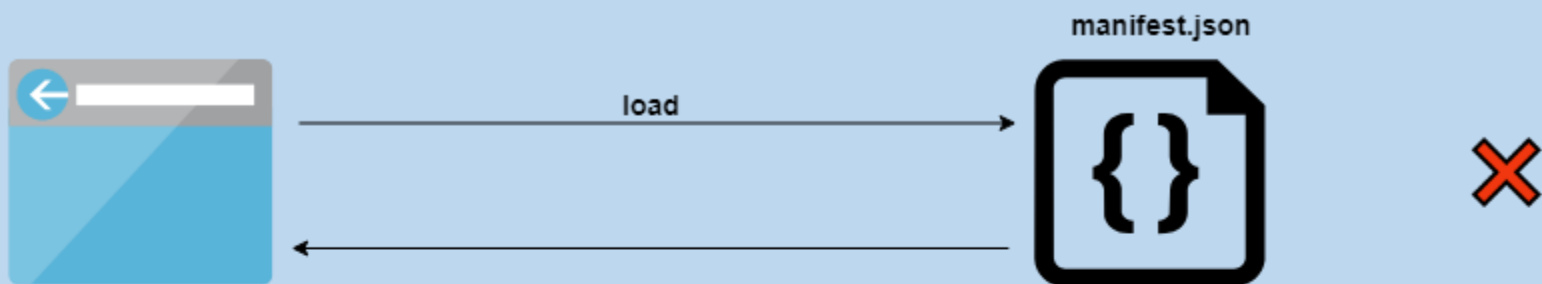
- Leverage `webpack-manifest-plugin`
- Provides consistently named entry point
- Allows aggressive caching and file hashing
- Integration apps consume manifest, inject script & style tags

```
{  
  "app.css": "/css/app.a094a6f6.css",  
  "app.js": "/js/app.c602be9d.js",  
  "chunk-vendors.css": "/css/chunk-vendors.719cc06f.css",  
  "chunk-vendors.js": "/js/chunk-vendors.a8c4a308.js"  
}
```

CLIENT

CDN

CACHE  
POLICY



# Added Benefits

- Vue.js work can be completed in isolation
- Dependency and framework encapsulation



*“Talk is cheap. Show me the code.”*

- Linus Torvalds

# Does it work in production?

The screenshot displays a web application interface for financial management. On the left is a dark sidebar with a navigation menu containing categories like DASHBOARD, PROFILES, SCREEN, PORTFOLIO ANALYSIS, and REPORTING. The main content area features a search bar at the top with the text 'Search for a Product' and a search input field. Below this is a section titled 'What would you like to do?' containing a list of interactive buttons such as 'Perform manager due diligence', 'Research and select new managers', and 'Monitor manager performance'. At the bottom of the main area is a link that says 'Still need some help?'. On the right side, there is a panel titled 'Alert Notifications' showing a list of 21 alerts. Each alert entry includes a title (e.g., 'Quartile change for Emerging Markets Equity'), a timestamp (e.g., '20 days ago'), and a status icon (a checkmark inside a circle).

# Questions?

