

React + PostMessage()

@CaraKuei



Cara Kuei

Frontend Engineer @



We are real.

(No)







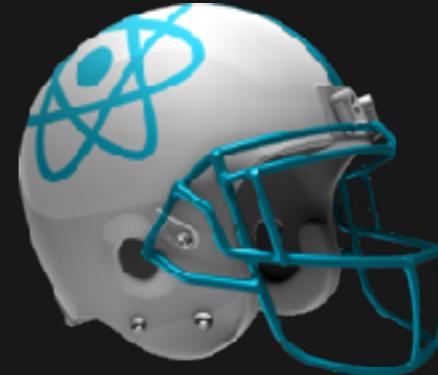


N



A





TL;DR: PostMessage

```
otherWindow.postMessage(message, targetOrigin);  
window.addEventListener("message", this.receiveMessage, false);
```

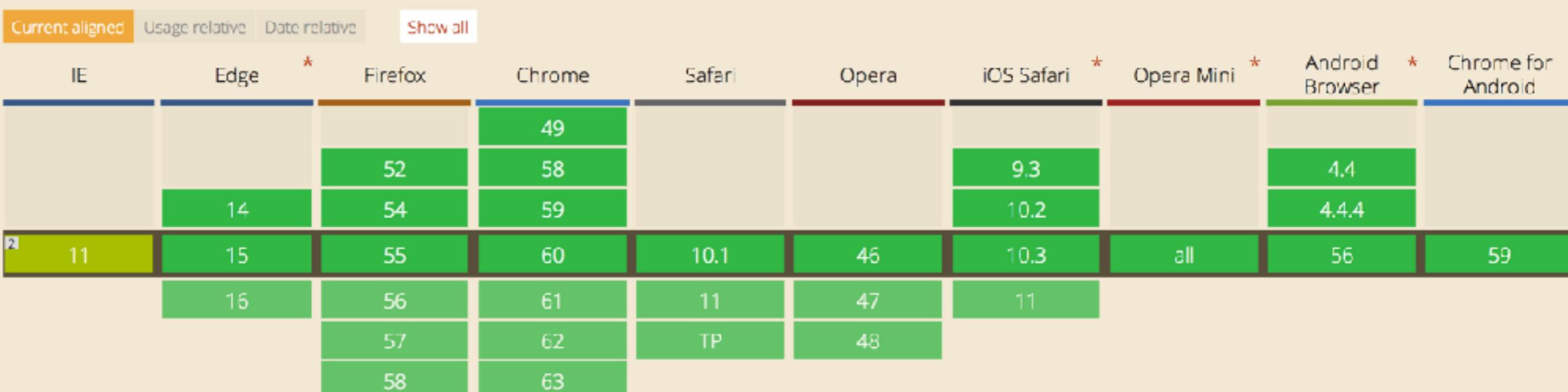
```
window.addEventListener("message", (event) => {console.log(event)}, false);
window.postMessage(
  "Are you ready for football?", // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

Cross-document messaging - LS

Global

93.52% + 4.19% = 97.71%

Method of sending information from a page on one domain to a page on a different one (using postMessage)



Notes

Sub-features (1)

Known Issues (3)

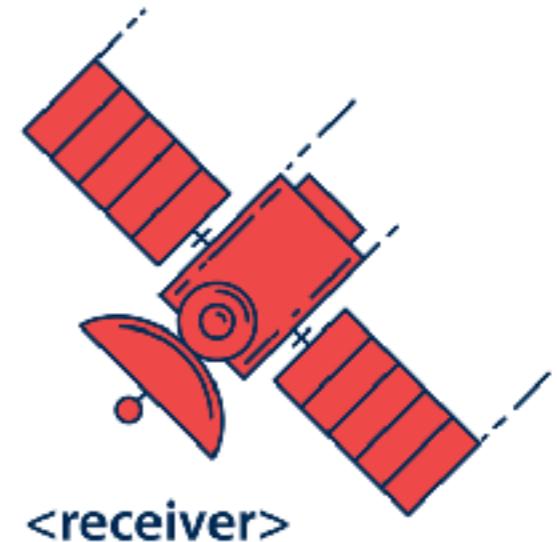
Resources (7)

Feedback

² Partial support refers to limitations in certain conditions.

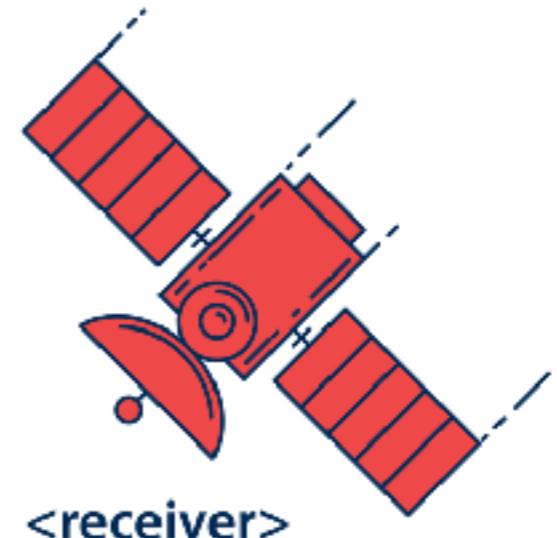
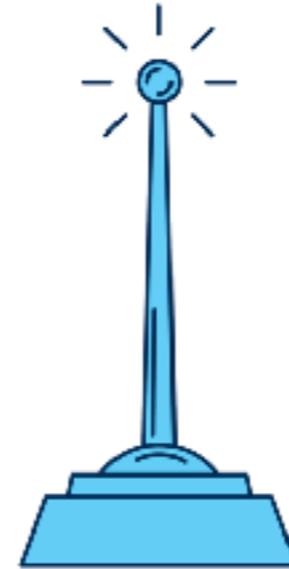
```
<parentComponent>  
  onClick()  
    <childComponent/>  
</parentComponent>
```

<sender>



```
<parentComponent>  
  onClick()  
    <childComponent/>  
</parentComponent>
```

<sender>



TL;DR: React

JK

IRL



Iterative migration
Simplify UX on recursive workflow

Background

NFL | Sign In Register | **HOME** | **SEARCH** | **LOGOUT**

FANTASY NEWS VIDEO SCORES SCHEDULE STANDINGS STATS WATCH GAMES TICKETS SHOP

SEARCH THIS SITE SEARCH NFL.COM

STAR PLAYERS ON THE BUBBLE

BOLD NFL PREDICTIONS

2017 NFL DRAFT

2017 NFL Training Camps & Locations

APPROVAL

2017 Fantasy Football Projections: Quarterbacks

APPROVAL

Dallas' Five Young Stars to Watch in 2017

APPROVAL

Predicting the Cowboys 2017 Pending Cardinals 2017 record

APPROVAL

32-year-old Fitzgerald makes NFL's oldest active receiver

APPROVAL

000 records with a young career ahead

APPROVAL

Brady Wins Top Back Plays

APPROVAL

Braves' Williams coaches his son, Brantley Price

APPROVAL

Unlikely QB: Possible guys who could be selected

APPROVAL

Get to get back to the Dallas Cowboys' connected body of work to begin year 5. See Part 1, and start with each part 4!

G

NEWS SCORES SCHEDULE STANDINGS VIDEO FANTASY

RIO GATHERS GOES UP AND GRABE TD

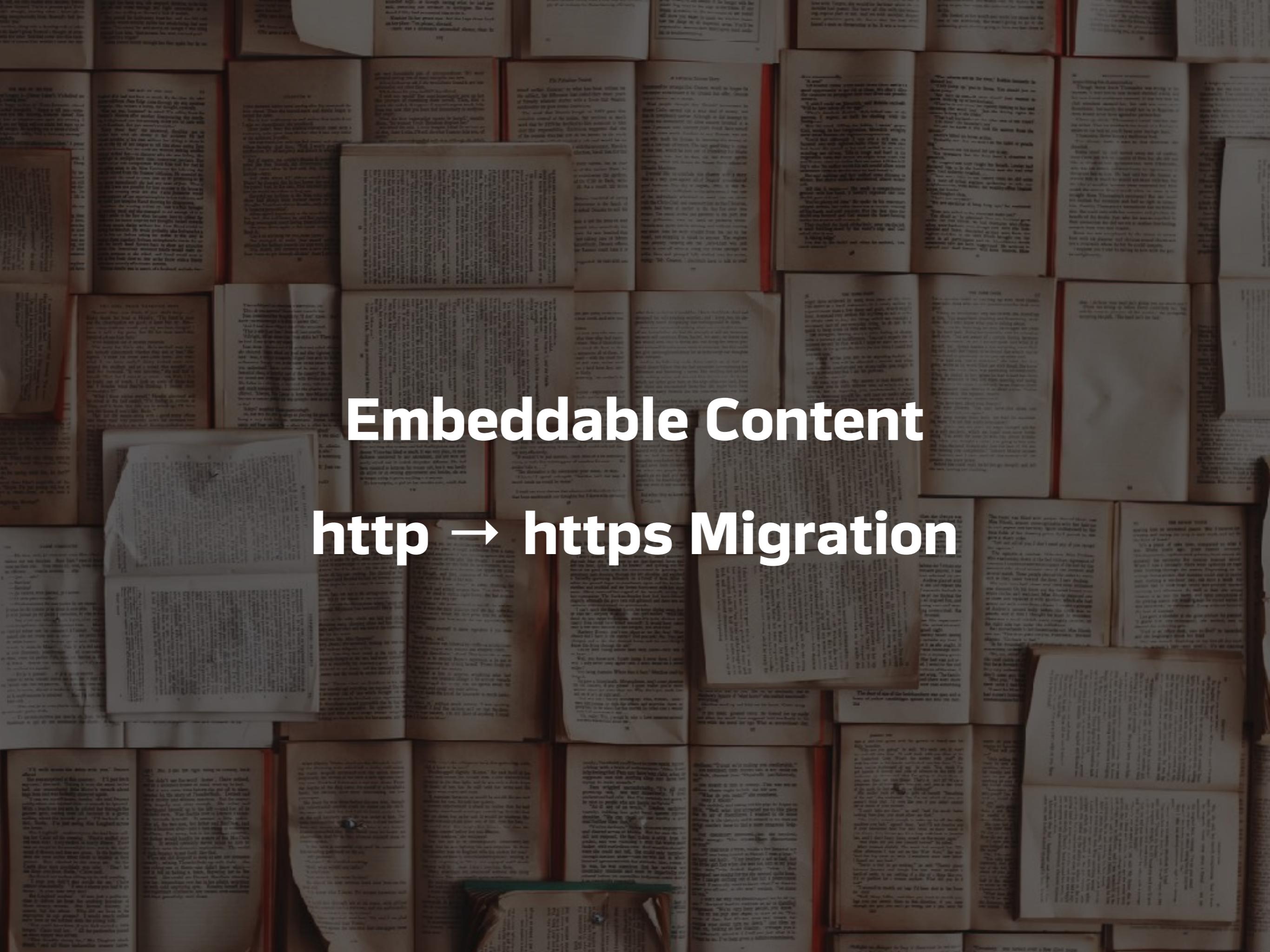
WHAT'S TRENDING

HALL OF FAME

TRAINING CAMPS

Getting better with age

Legacy vs New



Embeddable Content

http → https Migration

Embeddable Content

NYG 0-0	BAL 31	BUF 16	TB 12	MIN 13	CAR 27	KC 30	IND 19	DAL 24	NYJ 6	GB 21	NE 23	LA 24	DEN 25
CLE 0-0	MIA 7	PHI 20	JAX 8	SEA 20	TEN 34	CIN 12	DAL 24	DET 16	WAS 17	HOU 27	OAK 21	SF 21	FINAL
MON 8:00 PM ET	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL	FINAL



NEWS SCORES SCHEDULE STANDINGS VIDEO FANTASY ...

TICKETS SHOP TEAMS



News Home Transactions CFB 24/7 Features Podcasts Next Gen Stats



FANTASY DRAFT KIT

PRESENTED BY DIRECTV

15 players to target in fantasy football drafts in 2017



By Alex Gelhar
Fantasy writer/editor

Published: Aug. 14, 2017 at 08:03 p.m. Updated: Aug. 17, 2017 at 11:53 a.m.



◀ PREVIOUS ARTICLE

NEXT ARTICLE ▶

LATEST

[Rank's 11 Sleepers for 2017 fantasy football drafts](#)

[Streaming Week 1 fantasy football quarterback starts](#)

[Safest picks by round in 2017 fantasy football drafts](#)

[Making sense of the Patriots fantasy running backs](#)

[Rallest picks by round in 2017 fantasy football drafts](#)

[Don't fear reaching for Doug Baldwin, Jamison Crowder](#)

[15 players to target in fantasy football drafts in 2017](#)

[Fantasy football takeaways from preseason Week 1](#)

[Sammy Watkins, Jordan Matthews fantasy fallout](#)

[Ezekiel Elliott fantasy draft value falls with suspension](#)

[MORE POSTS](#)

SUBSCRIBE

[PLAYERS TO TARGET BY ROUND](#) [PLAYERS TO REACH FOR](#) [SAFEST PICKS BY ROUND](#)

Fantasy drafts are rapidly approaching (or are already underway). That means one of the most frequent questions asked to us analysts is "what round should I take PLAYER X" in? This is hard to say, because no two leagues will draft the same way, as they'll be filled with a different makeup of unique human beings with their own valuations of players. Some leagues are full of people who study fantasy religiously all summer. Others are made up of casual coworkers who will draft entirely based off the rankings in the draft room.



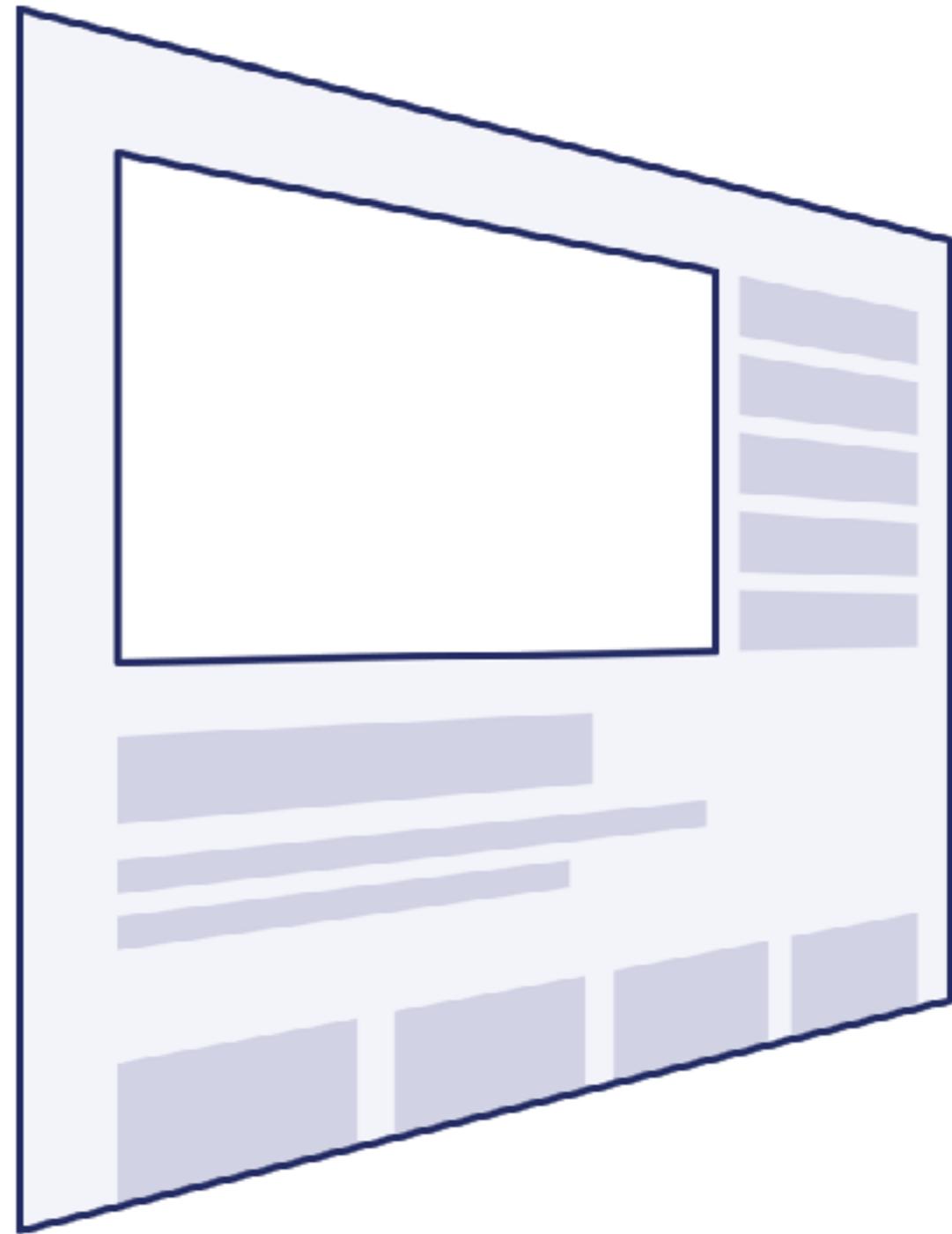
DIRECTV STARTS AT
\$60/mo

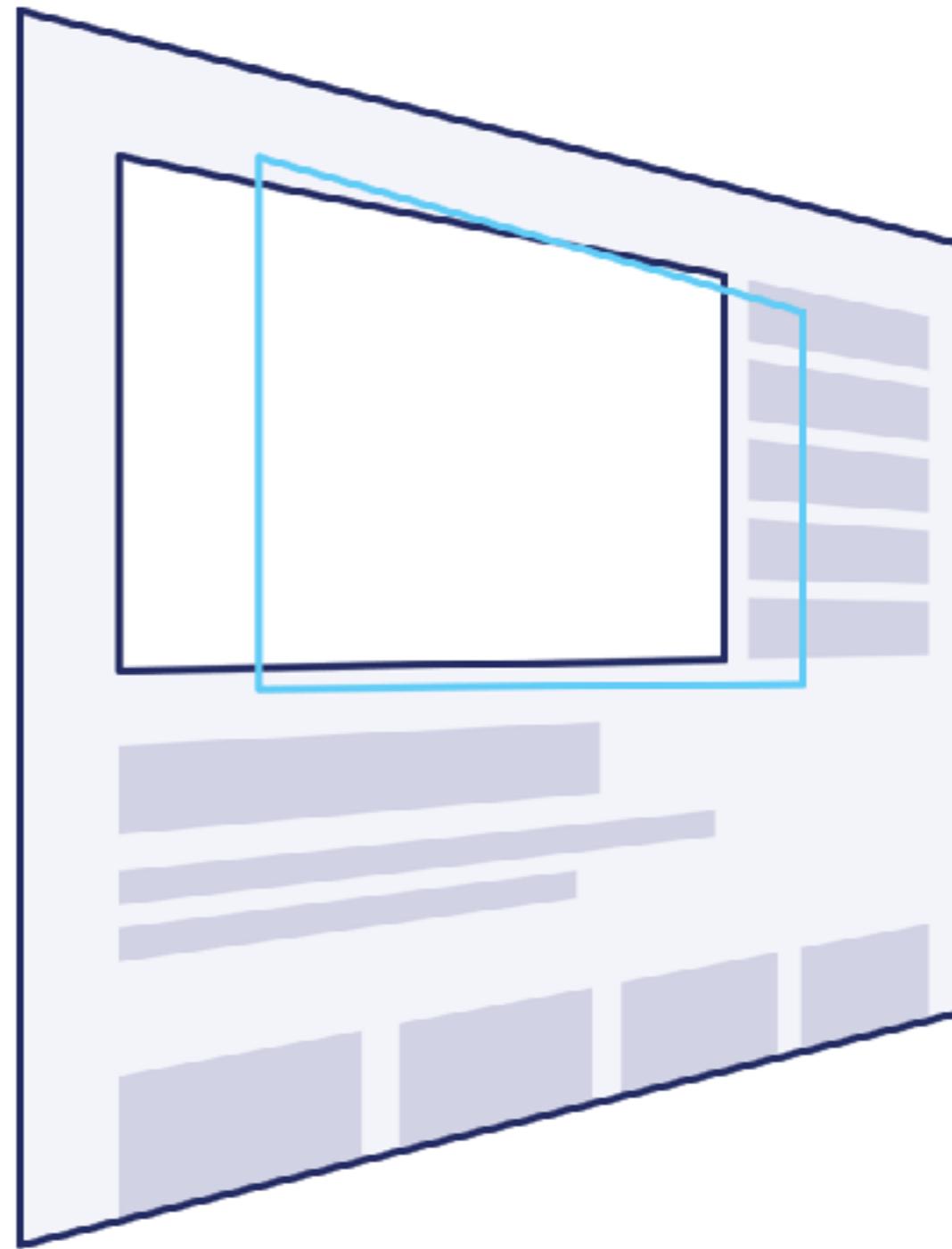
plus taxes

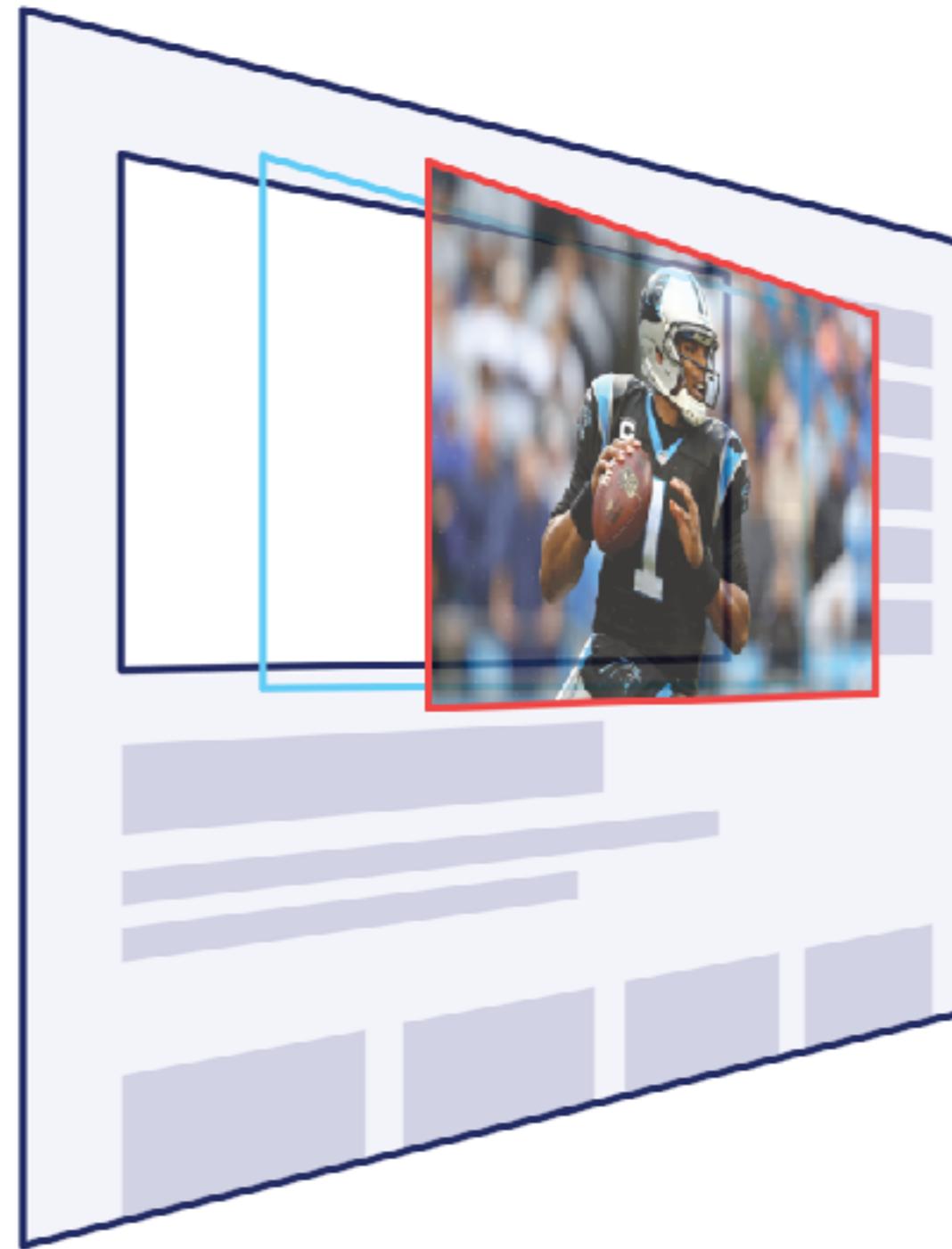
DIRECTV® SUNDAY TICKET® sub-serviced via satellite at then-prevailing rate (currently \$624.99/mo) unless cancelled prior to start of season.

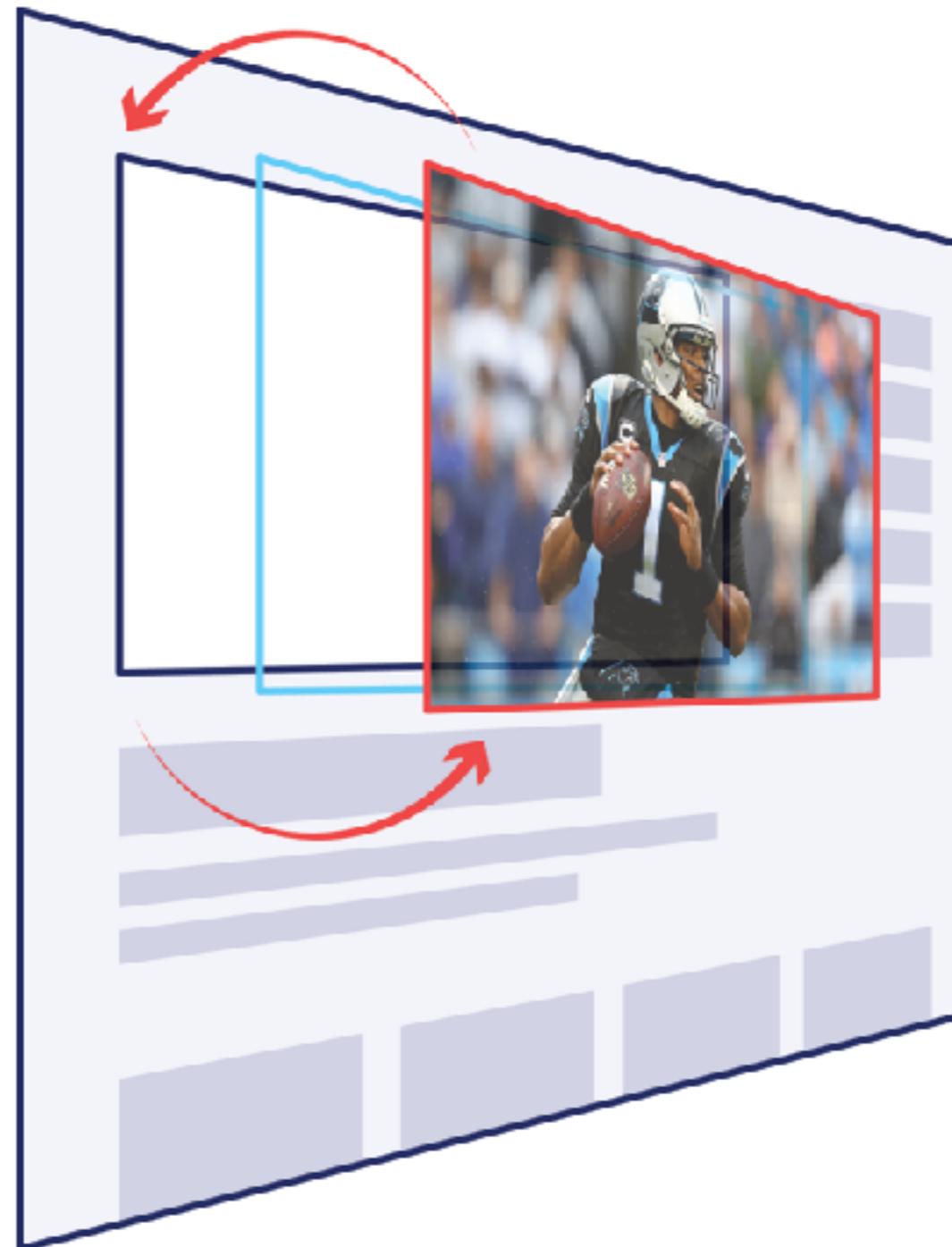
DIRECTV® SUNDAY TICKET® is a registered trademark of DIRECTV, Inc. © 2017 DIRECTV, Inc. All rights reserved. Regional Sports Networks. Local to \$7.50/mo, and taxes and fees apply. Prices higher in first year. Click for details.

Get DIRECTV









http → https

BAL	28	BUF	9	TB	12	MIN	0-0	CAR	0-0	KC	0-0	IND	0-0	NYJ	0-0	GB	0-0	NE	0-0	LA	
MIA	7	PHI	13	JAX	0	SEA	0-0	TEN	0-0	CIN	0-0	DAL	0-0	DET	0-0	PIT	0-0	HOU	0-0	OAK	
4TH • 08:32		1ST • 12:38		HALFTIME		FRI 10:00 PM ET		SAT 3:00 PM ET		SAT 7:00 PM ET		SAT 7:00 PM ET		SAT 7:30 PM ET		SAT 7:30 PM ET		SAT 8:00 PM ET		SAT 10:00 PM ET	



NEWS SCORES SCHEDULE STANDINGS VIDEO FANTASY ...

TICKETS SHOP TEAMS

GAME CENTER

Buffalo Bills Watch Listen

Philadelphia Eagles Listen Watch

LAST PLAY • 12:00 Q4
D.Evans
 (Shotgun) D.Evans pass incomplete short right to S.Oliver.

PLAY FANTASY FOOTBALL FOR FREE AT NFL.COM/FANTASY

◀ 2ND & 10 ▶

TRACK

ANALYZE

DISCUSS

VIDEOS

CURRENT DRIVE

LAST SCORE

BAL	28	BUF	9	TB	12	MIN	0-0	CAR	0-0	KC	0-0	IND	0-0	NYJ	0-0	GB	0-0	NE	0-0	LA	
MIA	7	PHI	13	JAX	0	SEA	0-0	TEN	0-0	CIN	0-0	DAL	0-0	DET	0-0	PIT	0-0	HOU	0-0	OAK	
4TH • 08:32		1ST • 12:38		HALFTIME		FRI 10:00 PM ET		SAT 3:00 PM ET		SAT 7:00 PM ET		SAT 7:00 PM ET		SAT 7:30 PM ET		SAT 7:30 PM ET		SAT 8:00 PM ET		SAT 10:00 PM ET	



NEWS SCORES SCHEDULE STANDINGS VIDEO FANTASY ...

SEARCH TICKETS SHOP TEAMS



GAME CENTER

Buffalo Bills
Watch Listen

9

2nd & 10 11:57 PHI 20

	1	2	3	4	OT
BUF	0	6	3	0	
PHI	3	10	0	0	

13



Philadelphia
Eagles

Listen

Watch

LAST PLAY • 12:00 Q4

D.Evans
(Shotgun) D.Evans pass incomplete short right to S.Oliver.

PLAY FANTASY FOOTBALL FOR FREE AT NFL.COM/FANTASY



Like Share

TRACK

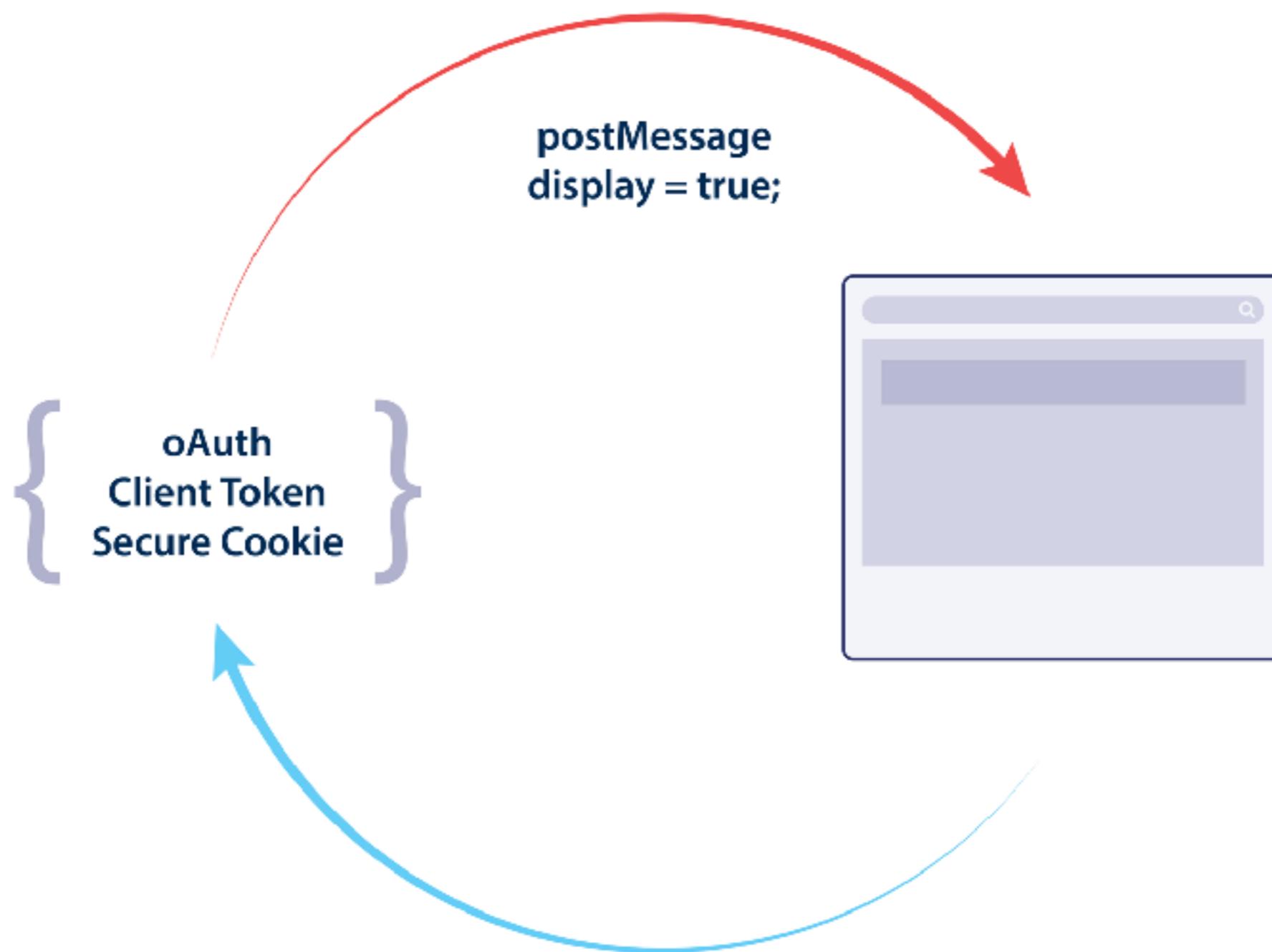
ANALYZE

DISCUSS

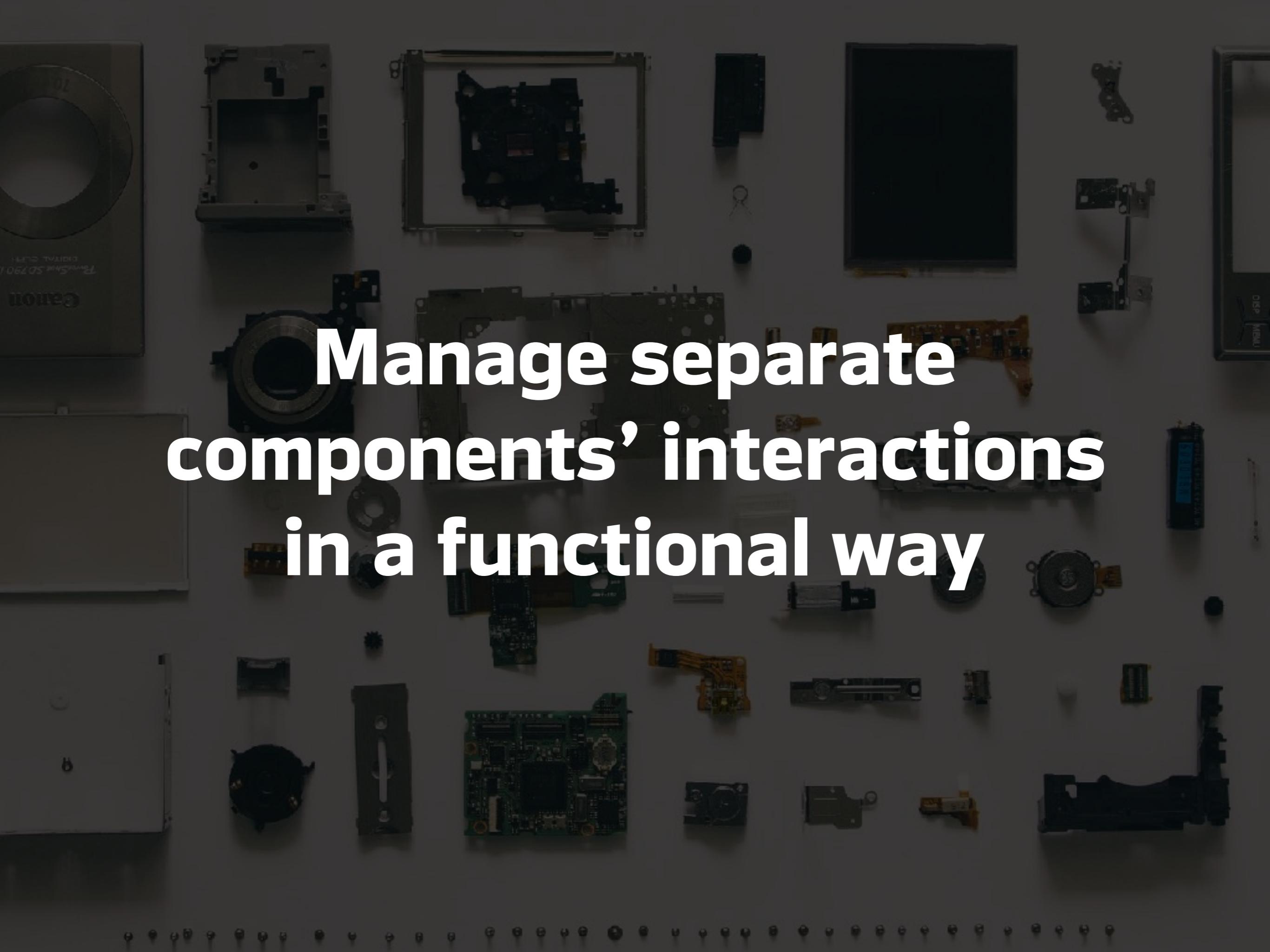
VIDEOS

CURRENT DRIVE

LAST SCORE



Simplify UX



**Manage separate
components' interactions
in a functional way**

*** Required**

Name *

 0

City

 0

Country

 0

Postal Code

State

 0

Street

 0

Title

 0

* Required

Name *

 0

Metrics

Super Bowl - Page Navigation x ▾  

Details

id: 848d601b-f86c-4dbd-bb82-0443023d1d9f
name: Super Bowl - Page Navigation
eventName: clickAction
linkModule: super bowl header click
linkName: null
linkType: header

Target

 0

Text *

 0

Url *

 0

This image is a promotional advertisement for NFL Game Pass. It features two Dallas Cowboys players in the foreground: Ezekiel Elliott, wearing jersey number 21, and Dak Prescott, wearing jersey number 4. Both players are shown in mid-air, performing a celebratory 'dab' or 'tuck' move. The background is a blurred football field at night.

The central text reads "NFL GAME>PASS" in a large, bold font. Below it is the tagline "Watch More Football". A blue button labeled "READ MORE" is positioned in the lower center. At the bottom, the text "Replay Every Game of the Season" is flanked by arrows pointing towards the "READ MORE" button.

Annotations with arrows point to specific elements:

- "Logo" points to the NFL shield logo located between the two players.
- "Image" points to the top right corner of the advertisement area.
- "Title" points to the word "GAME>PASS" in the main headline.
- "Subtitle" points to the text "Replay Every Game of the Season" below the main headline.
- "Button" points to the blue "READ MORE" button at the bottom center.

NFL CMS - Component Builder / instances

Secure | https://cms.nfl.com/admin/component-builder/instances/upsert/Hero/29712198-9932-4b11-bb2b-e915bcf9ff48

Brochure Builder Component Builder Notifications Polls SDET Users Videos caraqaye

Component Builder / Instances / Hero / _CaraReactRally

PUBLISHED

Meta Fields

Domain Fields

* Required

Name *

_CaraReactRally 17

Logo

_Test_Cara_B 16

Details

Image

NDC - Homepage - Hero Image - Dwight Clark The Catch - 7/11/17 16

Details

Title

Coming up clutch 16

Subtitle

From the Hail Mary to The Catch, these are the 10 most clutch touchdowns of all time 84

Call To Actions

Create



NFL CMS - Component Builder / Ins X

Secure | https://cms.wildcat.nfl.com/admin/component-builder/instances/upsert/Hero/29712f98-9932-4bf1-bb2b-e915bcf9ff48

Brochure Builder Component Builder Notifications Polls SDET Users Videos caraqaye

Component Builder / Instances / Hero / _Cara_React_Rally

PUBLISHED

Meta Fields

Domain Fields

* Required

Name * 17

Logo x + ⌂

Details

id: 190a57b3-9467-42ff-835b-0be531ed5743

name: _Test_Cara_5

alt: React Rally

height: null

src: null

width: null

mediaSizes: 5cbaad62-edbf-4a26-a59d-c3fdc03d2d58

Image x + ⌂



postMessage()



Details

Title

Coming up clutch

16

Subtitle

From the Hail Mary to The Catch, these are the 10 most clutch touchdowns of all time

84

Call To Actions

Create



Footnote

0

Type

Select...



Videos

Create

Sources

Create

Preview Config

Select...



Save

Copy

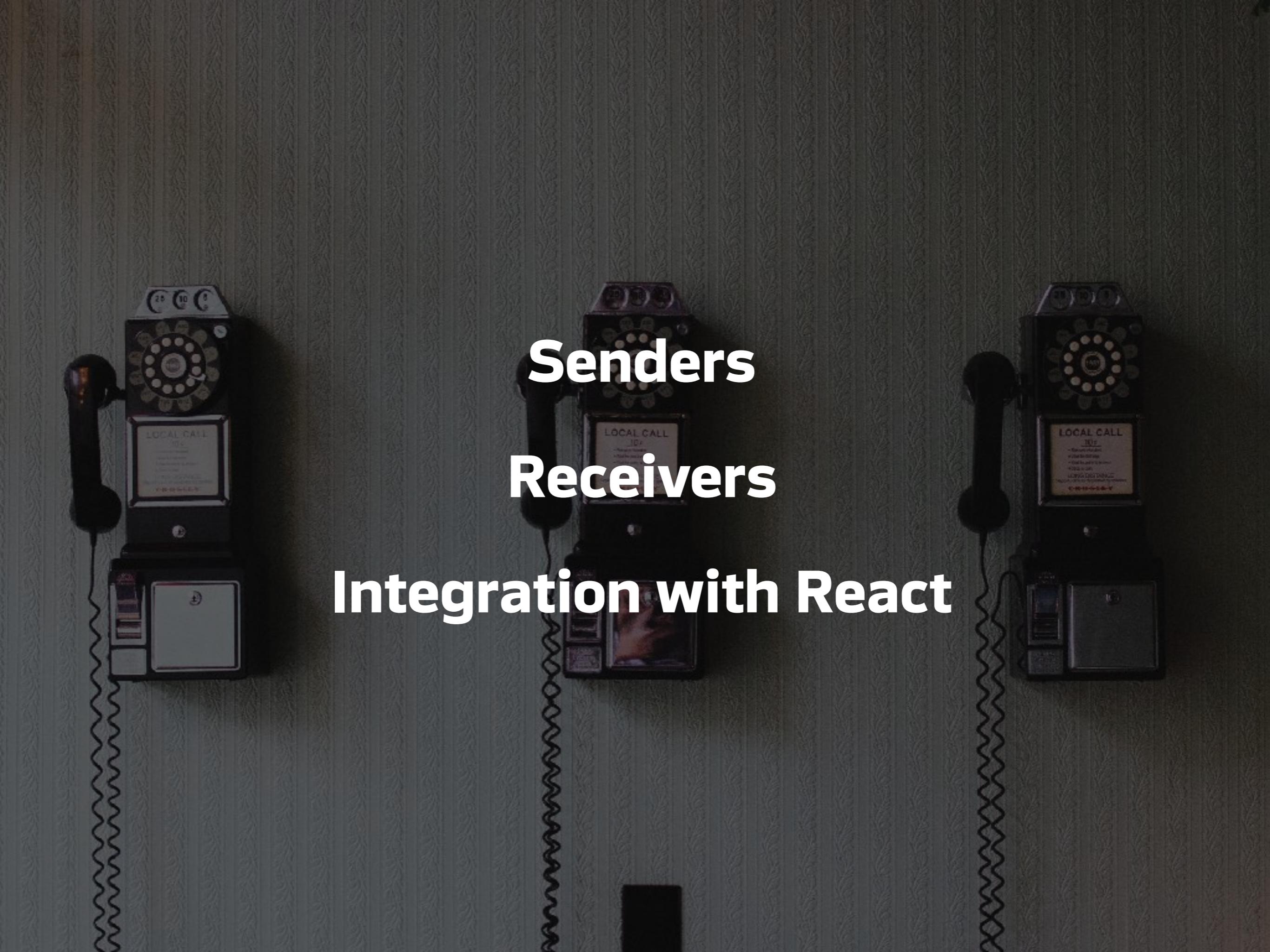
Cancel

Save and Publish





DIY: Use postMessage



Senders Receivers

Integration with React

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. At the top, there are icons for Refresh, Stop, Elements, Console (which is underlined), Audits, Sources, Network, and two more tabs. To the right of these are notification icons (yellow triangle with '5'), three vertical dots, and a close button. Below the tabs, there's a search bar with a magnifying glass icon, a 'top' dropdown, a 'message' filter input, and a 'Info only' dropdown showing '17 items hidden by filters'. A gear icon is also present. The main area displays a single line of code:

```
> window.addEventListener("message", (event) => {  
    console.log("message", event.data);  
});
```

```
> window.addEventListener("message", (event) => {  
    console.log("message", event.data);  
});
```

Sta

say I'm not a

```
window.postMessage(  
    "Are you ready for football?", // the message you are sending  
    "https://www.nfl.com" // the target origin  
);
```

Supported types

Object type	Notes
All primitive types	However not symbols
Boolean object	
String object	
Date	
RegExp	The <code>lastIndex</code> field is not preserved.
Blob	
File	
FileList	
ArrayBuffer	
ArrayBufferView	This basically means all typed arrays like <code>Int32Array</code> etc.
ImageData	
Array	
Object	This just includes plain objects (e.g. from object literals)
Map	
Set	

```
window.postMessage(  
    "Are you ready for football?", // the message you are sending  
    "https://www.nfl.com" // the target origin  
);
```

```
window.addEventListener("message", this.onReceiveMessage, false);

const onReceiveMessage = (event: Object) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }
  const {data} = event;
  console.log(data); // "Are you ready for football?"
};
```

```
window.addEventListener("message", this.onReceiveMessage, false);

const onReceiveMessage = (event: Object) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }
  const {data} = event;
  console.log(data); // "Are you ready for football?"
};
```

Have to be more specific!

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  ${JSON.stringify(message)}, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  ${JSON.stringify(message)}, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  ${JSON.stringify(message)}, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  ${JSON.stringify(message)}, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

window.addEventListener("message", this.onReceiveMessage, false);

const onReceiveMessage = (event: Event) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }

  const {data}: {data: string} = event;
  const message: MessageType = JSON.parse(data);
  const {type} = message;
  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
};
```

```
/* @flow */

window.addEventListener("message", this.onReceiveMessage, false);

const onReceiveMessage = (event: Event) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }

  const {data}: {data: string} = event;
  const message: MessageType = JSON.parse(data);
  const {type} = message;
  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
};


```

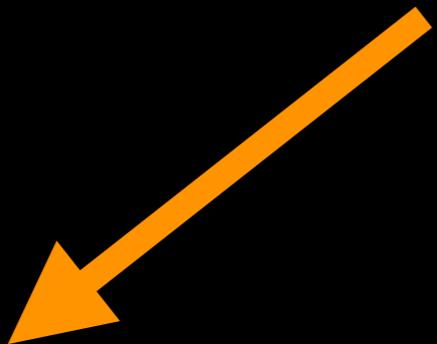
```
/* @flow */

window.addEventListener("message", this.onReceiveMessage, false);

const onReceiveMessage = (event: Event) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }

  const {data}: {data: string} = event;
  const message: MessageType = JSON.parse(data);
  const {type} = message;
  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
};
```

```
const type MessageType: {  
    data: {}, // just an object that probably means something in our system  
    type: string // event name that means something in our system  
};
```



```
const type MessageType: {  
    data: {}, // just an object that probably means something in our system  
    type: string // event name that means something in our system  
};
```

How do I know if this
message is for my
component?

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The top navigation bar includes 'Elements', 'Console' (which is underlined), 'Audits', 'Sources', 'Network', and other icons. A status bar on the left says 'Sta'. The main console area has a search bar with a magnifying glass icon and a dropdown menu. Below that, there's a filter bar with a 'message' input field and a 'Info only' dropdown. A message count of '5' is shown in the top right corner. The console log shows a single entry:

```
> window.addEventListener("message", (event) => {
    console.log("message", event.data);
});
```

say I'm not a

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  `demo${JSON.stringify(message)}`, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

const message: MessageType = {
  data: {
    id: "12345",
    "firstName": "Tom",
    "lastName": "Brady",
    "activeRole": "PLAYER",
  } // Data you care about
  type: "logging" // the action to take on the data
};

window.postMessage(
  `demo${JSON.stringify(message)}`, // the message you are sending
  "https://www.nfl.com" // the target origin
);
```

```
/* @flow */

window.addEventListener("message", this.onReceiveMessage, false);

const namespace = "demo";

const onReceiveMessage = (event: Event) => {
  const {data}: {data: string} = event;
  if (!data.includes(namespace)) {
    return;
  }
  const message: MessageType = JSON.parse(data.substr(namespace.length));
  const {type} = message;
  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
};
```

```
/* @low */

window.addEventListener("message", this.onReceiveMessage, false);

const namespace = "demo";

const onReceiveMessage = (event: Event) => {
  const {data}: {data: string} = event;
  if (!data.includes(namespace)) {
    return;
  }

  const message: MessageType = JSON.parse(data.substr(namespace.length));
  const {type} = message;
  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
}
```

What if you miss a
message?

Everyone needs to be
both **listener** and
Sender!

Hey, let me know when
you're ready.

S



Hey, let me know when
you're ready.

S

R

Hey! I'm ready :)



Hey, let me know when
you're ready.

S

R

Hey! I'm ready :)

Okay, here's the message.

S



Hey, let me know when
you're ready.

S

R

Hey! I'm ready :)

Okay, here's the message.

S

R

Alright. Let me do some...



Hey, let me know when
you're ready.

S

R

Hey! I'm ready :)

Okay, here's the message.

S

R

Alright. Let me do some...

R

Actually, I'm done. Here are
the results :)



Hey, let me know when
you're ready.

S

R

Hey! I'm ready :)

Okay, here's the message.

S

R

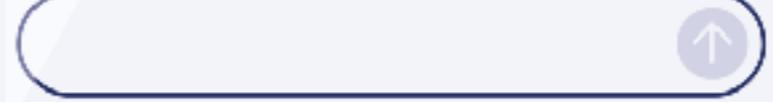
Alright. Let me do some...

R

Actually, I'm done. Here are
the results :)

S

Awesome, thanks! I'll update
myself with these results.



```
componentDidMount() {  
    window.addEventListener("message", this.receiveMessage, false);  
    this.setupPage();  
}  
}
```

```
componentWillUnmount() {  
    window.removeEventListener("message", this.receiveMessage);  
}  
}
```

```
/* @flow */

window.addEventListener("message", this.onReceiveMessage, false);

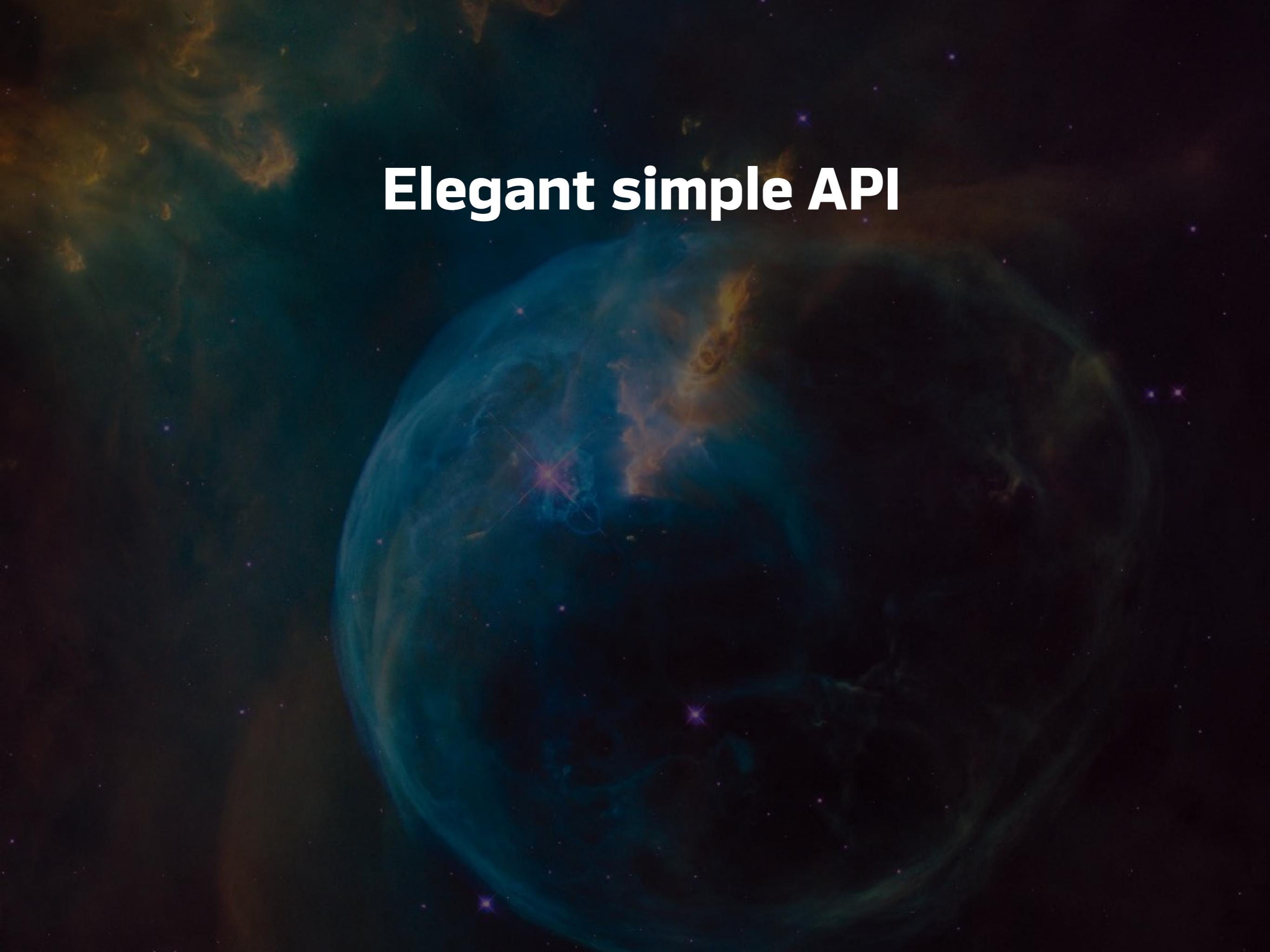
const onReceiveMessage = (event: Event) => {
  if (event.origin !== "https://www.nfl.com") {
    return;
  }

  const {data}: {data: string} = event;
  const message: MessageType = JSON.parse(data);
  const {type} = message;

  switch(type) {
    case "ready":
      // set up
    case "logging":
      const {data} = message;
      console.log(` ${data.firstName} ${data.lastName}`); // "Tom Brady"
  }
};


```

ICYMI



Elegant simple API

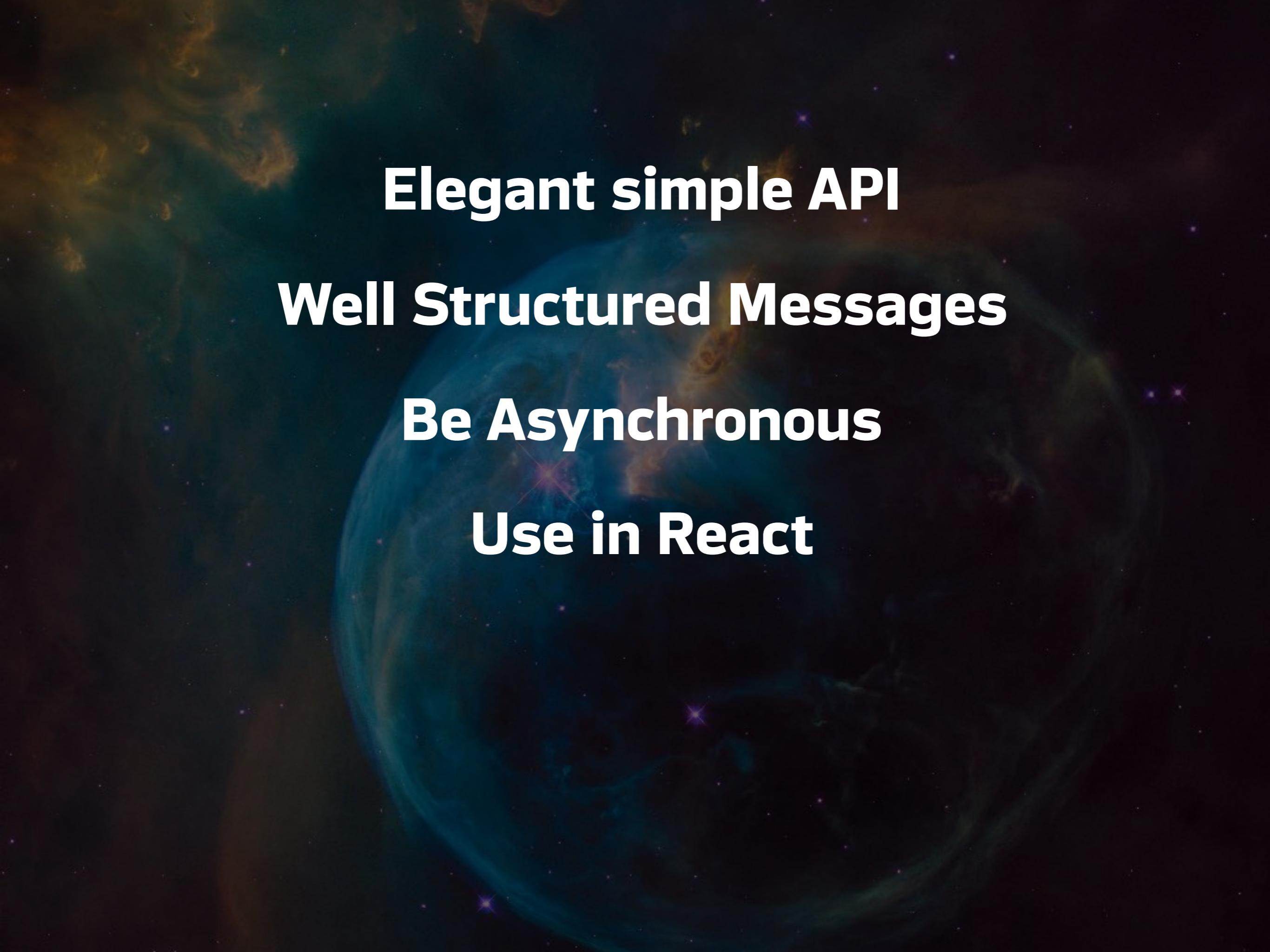
Elegant simple API

Well Structured Messages

Elegant simple API

Well Structured Messages

Be Asynchronous

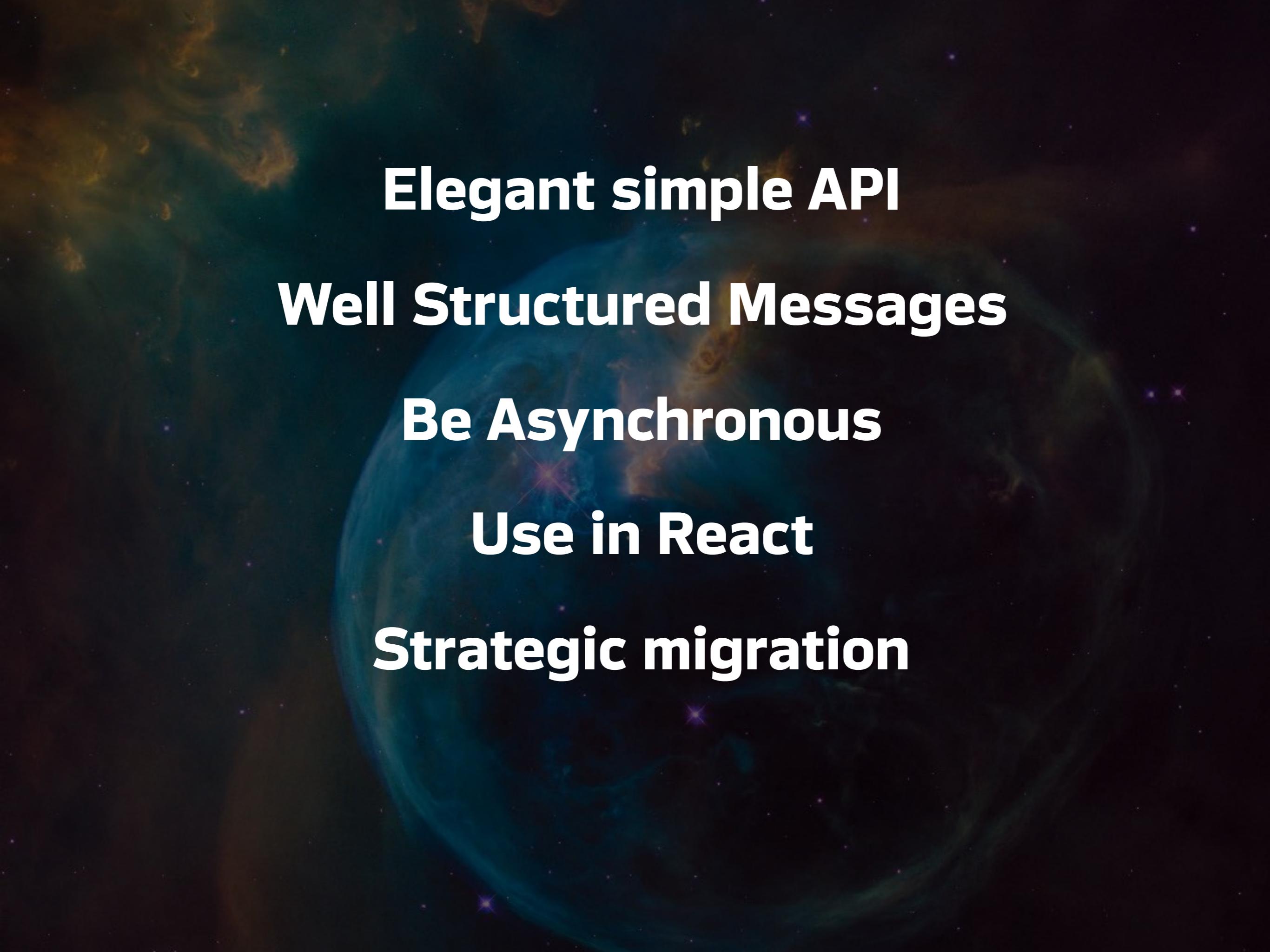


Elegant simple API

Well Structured Messages

Be Asynchronous

Use in React



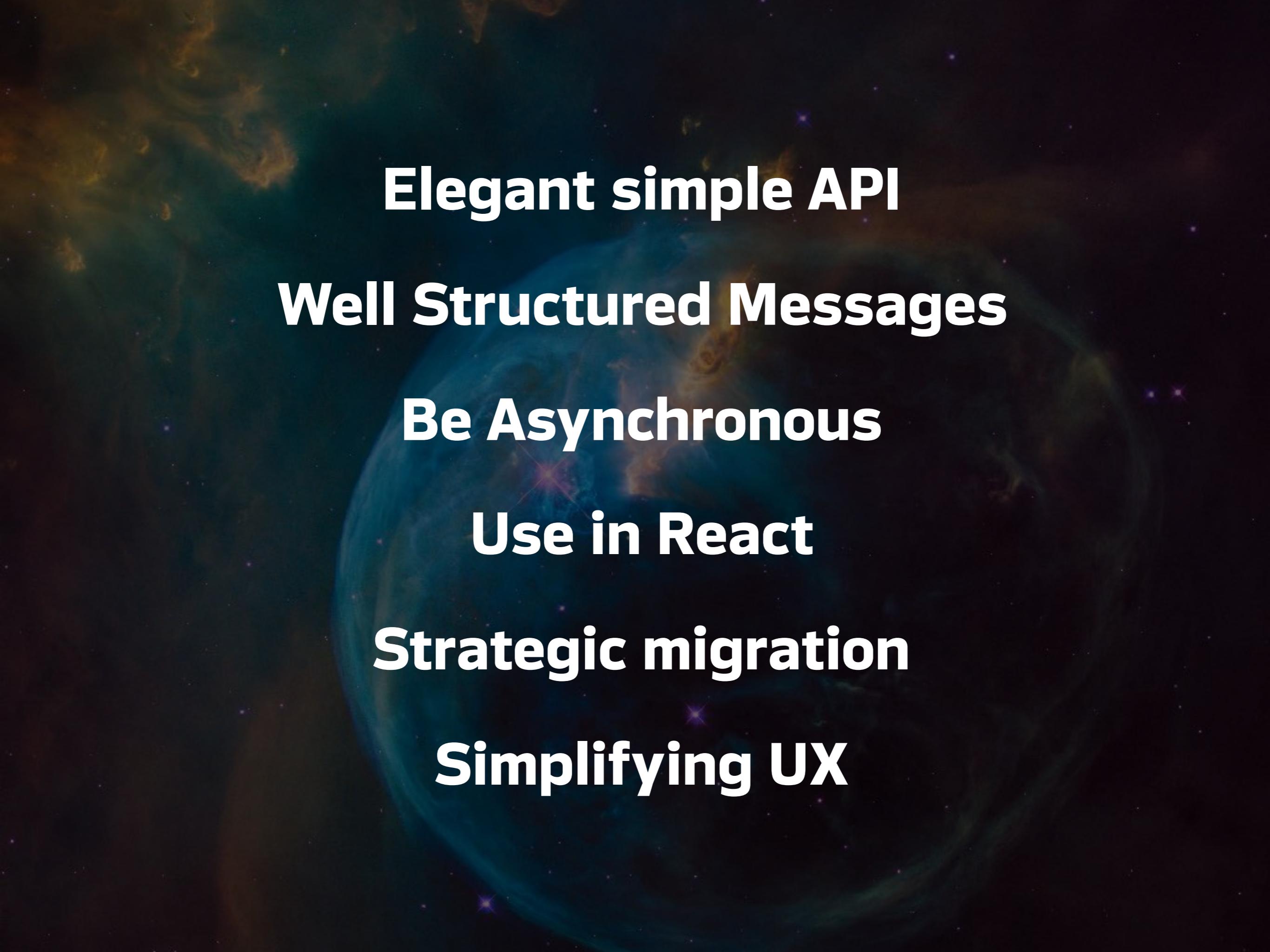
Elegant simple API

Well Structured Messages

Be Asynchronous

Use in React

Strategic migration



Elegant simple API

Well Structured Messages

Be Asynchronous

Use in React

Strategic migration

Simplifying UX

One more thing.



JK JK

kthxbai



@CaraKuei



@nflengineers