

# Stop Building APIs

Options for automating  
end-to-end communication.

**Masking Technology**

hello@masking.tech

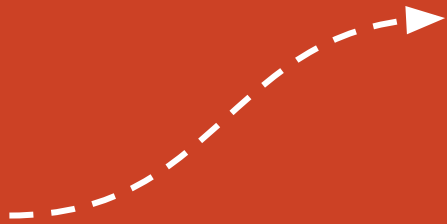
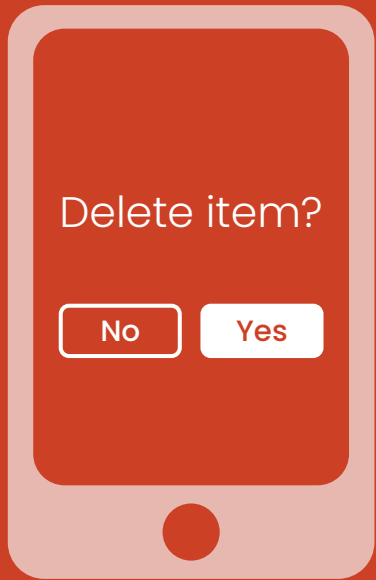
[linkedin.com/company/maskingtechnology](https://www.linkedin.com/company/maskingtechnology)



# Session breakdown



- 1
- 2
- 3
- 4



- 1
- 2
- 3
- 4

Frontend

Backend



UI



API request



API endpoint



Logic

- 1
- 2
- 3
- 4



Developer

Building and maintaining APIs takes a lot of time.

APIs for internal communication do not add any business value.



Architect

- 1
- 2
- 3
- 4





**tRPC**



**Remult**



**Tempo**

1

2

3

4

## tRPC → Typesafe API

```
// Server code
const appRouter = router({
  deleteItem: publicProcedure
    .input(z.object({ id: z.string() }))
    .query(async ({ input }) => { const id = input.id; /* ... */ })
});

// Client code
const trpc = createTRPCProxyClient<AppRouter>({
  links: [ httpBatchLink({ url: 'http://localhost:3000' }) ]
});

await trpc.deleteItem.query('1');
```



1

2

3

4

## Result → CRUD Framework

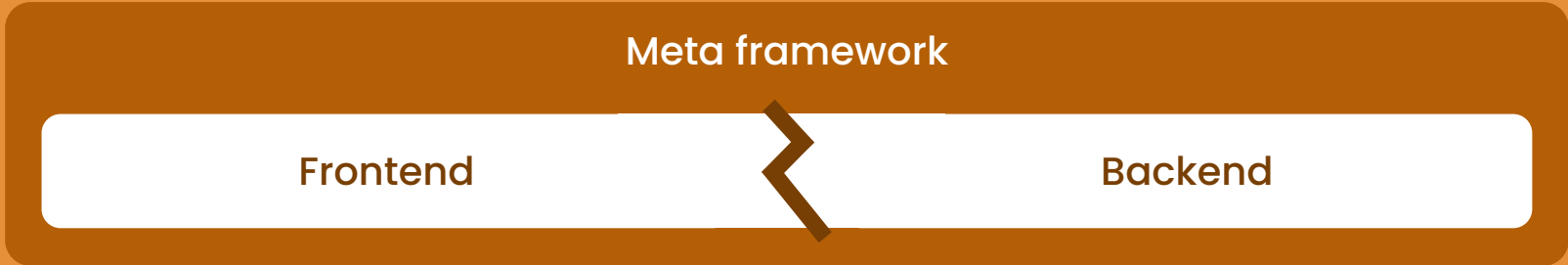
```
// Server code
@Entity('items', { allowApiCrud: true })
class Item { @Fields.uuid() id!: string; /* other fields */ }

app.use(resultExpress({
  entities: [Item]
}));

// Client code
const itemsRepo = result.repo(Item);

const item = await itemsRepo.findById('1');
await itemsRepo.delete(item);
```

- 1
- 2
- 3
- 4







## Next.js → Server Actions

```
async function deleteItem(id: string) {
  'use server';
  /* ... */
}

export default function DeleteItemDialog({id}: {id: string}) {
  return (
    <!-- ... -->
    <button onClick={() => startTransition(() => deleteItem(id))}>
      Yes
    </button>
  );
}
```

1

2

3

4

## SolidStart → Server Functions

```
const deleteItem = server$(async (id: string) => {
  /* ... */
});

export default function DeleteItemDialog({id}: {id: string}) {
  return (
    <!-- ... -->
    <button onClick={() => deleteItem(id)}>
      Yes
    </button>
  );
}
```



Framework

Mediator

- + Fully automated
- + Out-of-the-box
- Extra vendor lock-in

- + Better portability
- + Freedom of choice
- Partly automated

### tRPC

```
const appRouter = router({
  deleteItem: publicProcedure
    .input( /* ... */ )
    .query( /* ... */ )
});
```



### Next.js

```
async function deleteItem(id: string)
{
  'use server';
  /* ... */
}
```

Migrating to another solution requires code changes.



Developer

I don't have to build APIs and have end-to-end type safety!

Change is the only constant. I hope we made the right choices.



Architect





If the problem is in the code, why not take it out?

1

2

3

4

If it can't be in the code,  
then where to put it?



- 1
- 2
- 3
- 4

### DeleteItemDialog.ts

```
import delete from './deleteItem'; automation  
  
function DeleteItemDialog(/* ... */) {  
  /* onClick => delete(...) */  
}
```

### deleteItem.ts

```
async function deleteItem(id: string) {  
  // Delete item from database  
}  
  
export default deleteItem;
```

JavaScript Modules to the rescue.

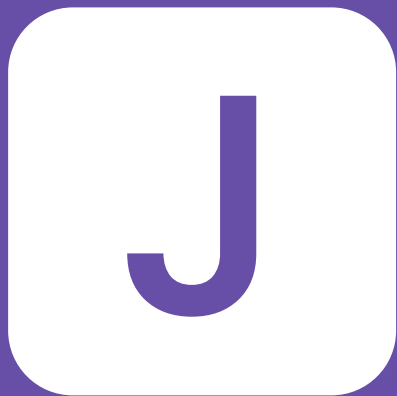


What runs where can be configured.

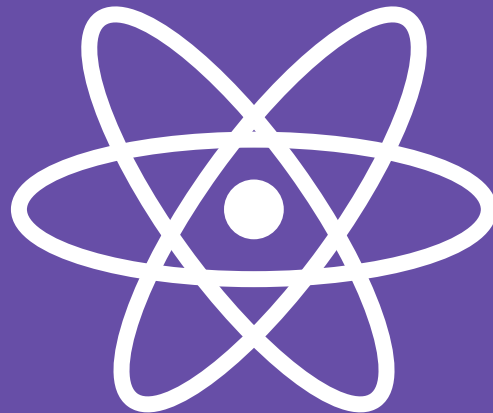
- 1
- 2
- 3
- 4



Environments can have different deployment models.



Jitar



React



Demo

```
> npm create jitar@latest
```

## Website



<https://jitar.dev>



## Github



[https://github.com/  
MaskingTechnology/  
jitar](https://github.com/MaskingTechnology/jitar)



# Thanks!



<https://masking.tech>

Bas Meeuwissen

[bas@masking.tech](mailto:bas@masking.tech)

[linkedin.com/in/basmeeuwissen](https://www.linkedin.com/in/basmeeuwissen)



Peter van Vliet

[peter@masking.tech](mailto:peter@masking.tech)

[linkedin.com/in/petervliet](https://www.linkedin.com/in/petervliet)

