SPECTEROPS

The Dog Ate My Homework…

(re)Building a PowerShell Module for the New BloodHound

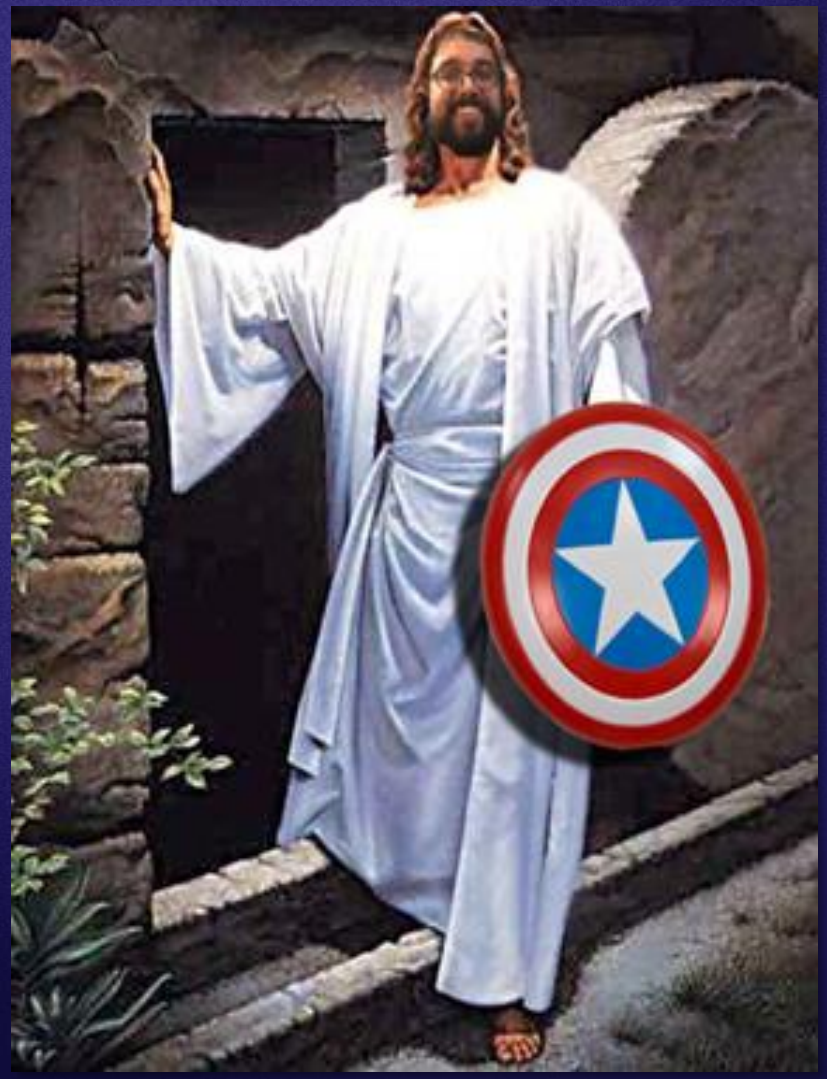
SadProcessor
SpecterOps

whoami

whoami

Some say

I love the Captain…



whoami

Some say

I dream in Graphs…

whoami

Others say

I'm just a PowerShell Bad Boy…

whoami

I can
neither confirm
nor deny
any of these
statements…

whoami

- SadProcessor – Walter Legowski
- Born FR / Home NL / Father x3
- ERNW / FalconForce / SpecterOps
- PowerShell all the Things!!
- Dog Whisperer Handbook
- CypherDog / WatchDog / ZipHound
  BHQlib / FalconHound

whoami

- SadProcessor – Walter Legowski
- Born FR / Home NL / Father x3
- ERNW / FalconForce / SpecterOps
- PowerShell all the Things!!
- Dog Whisperer Handbook
- CypherDog / WatchDog / ZipHound
  BHQlib / FalconHound

whoami

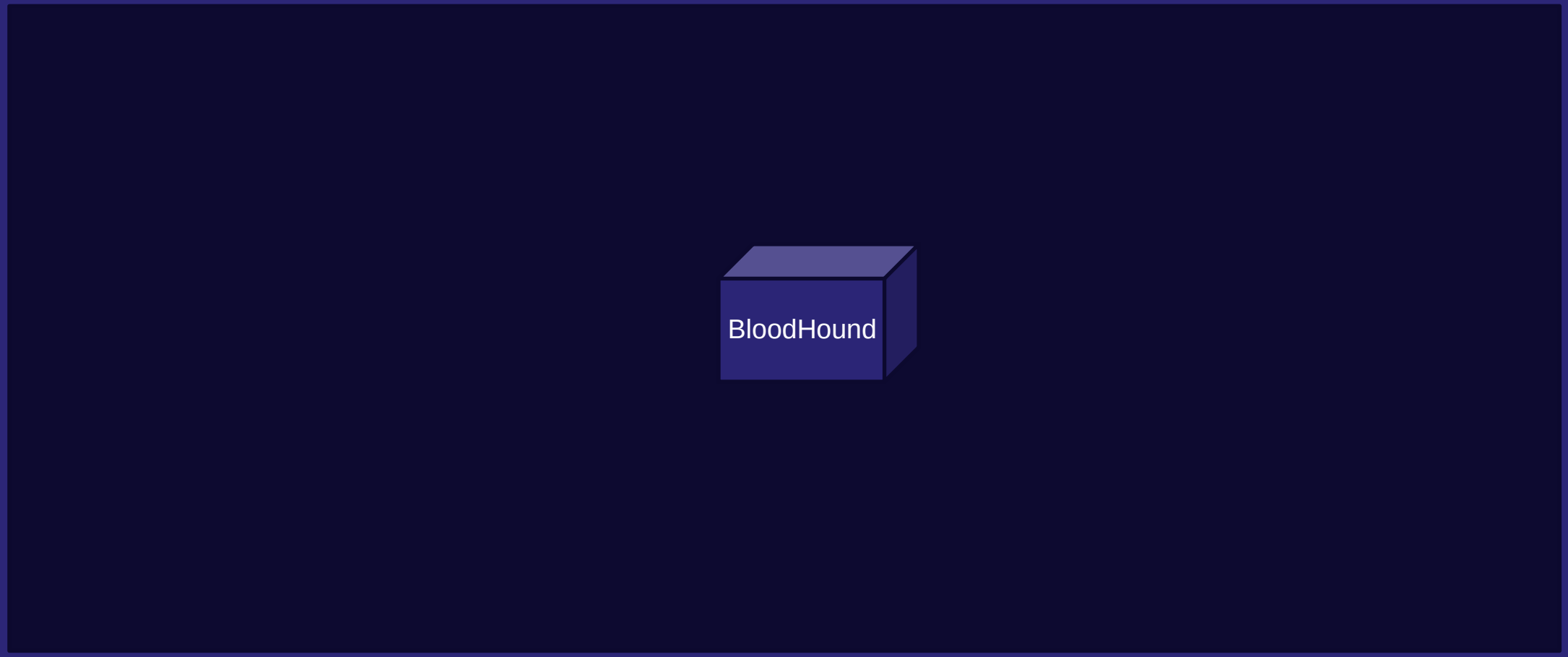
- SadProcessor – Walter Legowski
- Born FR / Home NL / Father x3
- ERNW / FalconForce / SpecterOps
- PowerShell all the Things!!
- Dog Whisperer Handbook
- PowerShell BloodHound Operator

Agenda

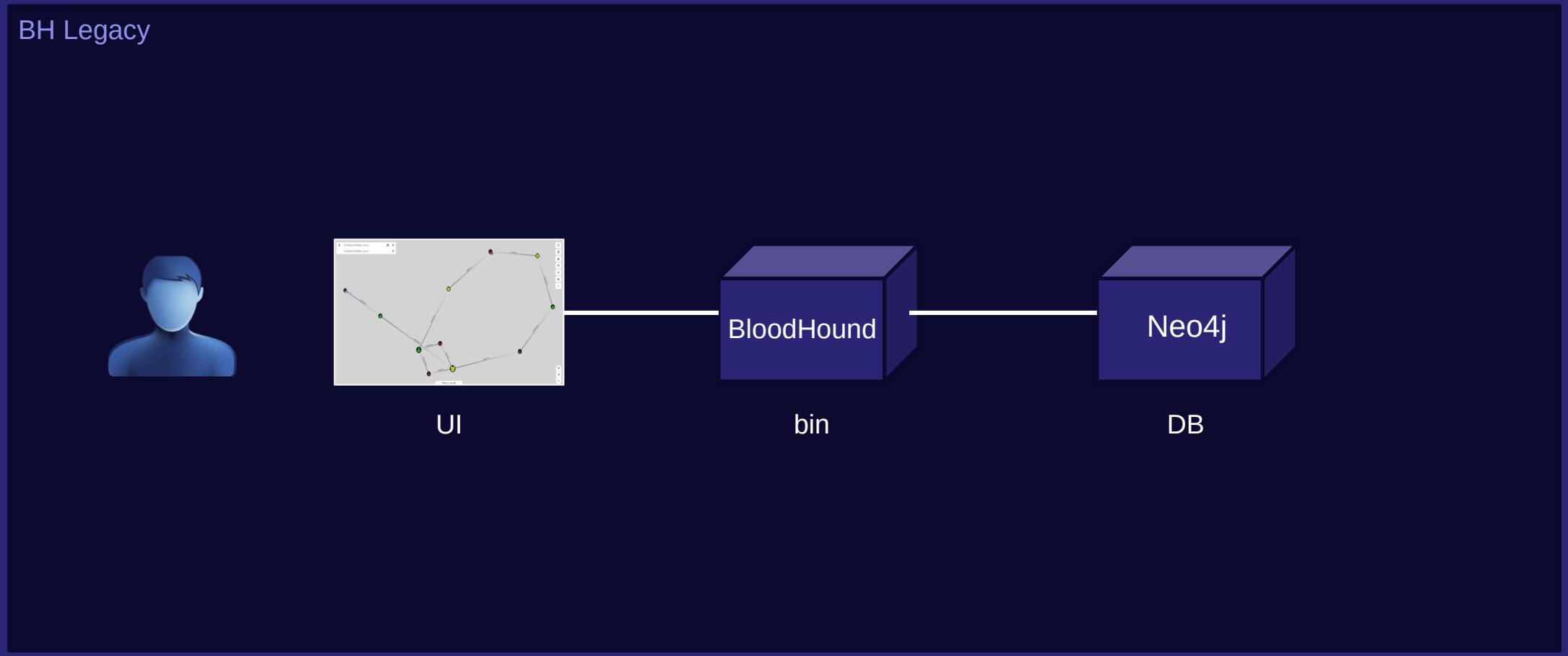
- Intro
- Old BloodHound vs. New BloodHound
- New BloodHound REST API
- PowerShell BloodHound Operator
- What Next…
- Q&A

Old BloodHound vs. New BloodHound

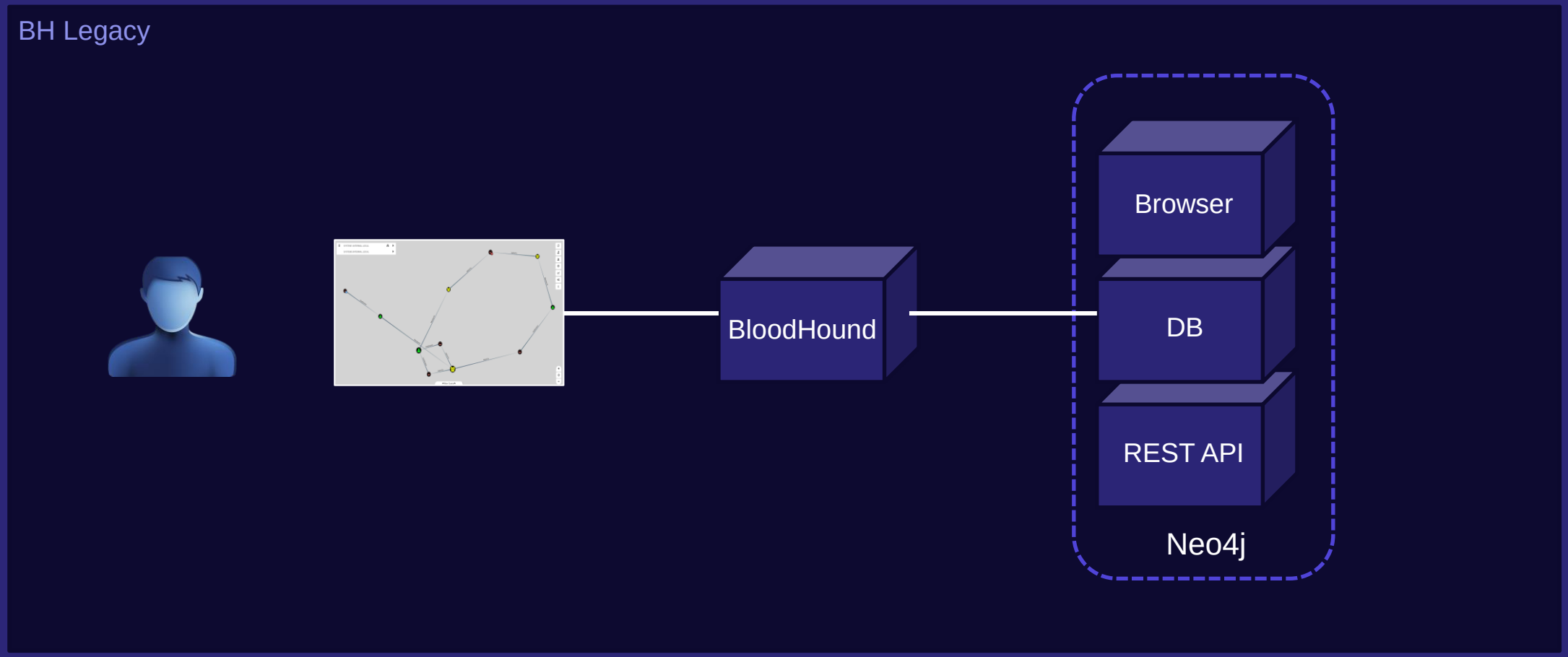
Old vs. New – Components



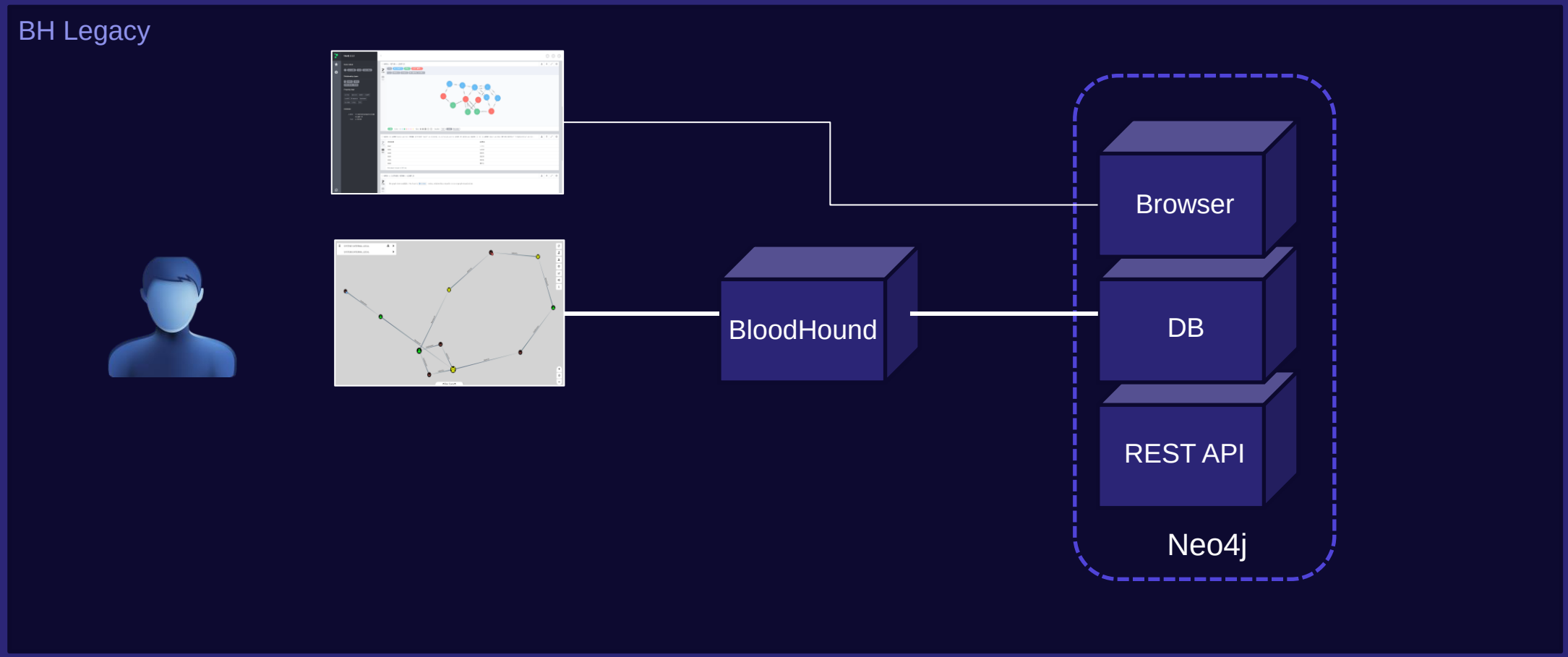
Old vs. New – Components: Old BloodHound



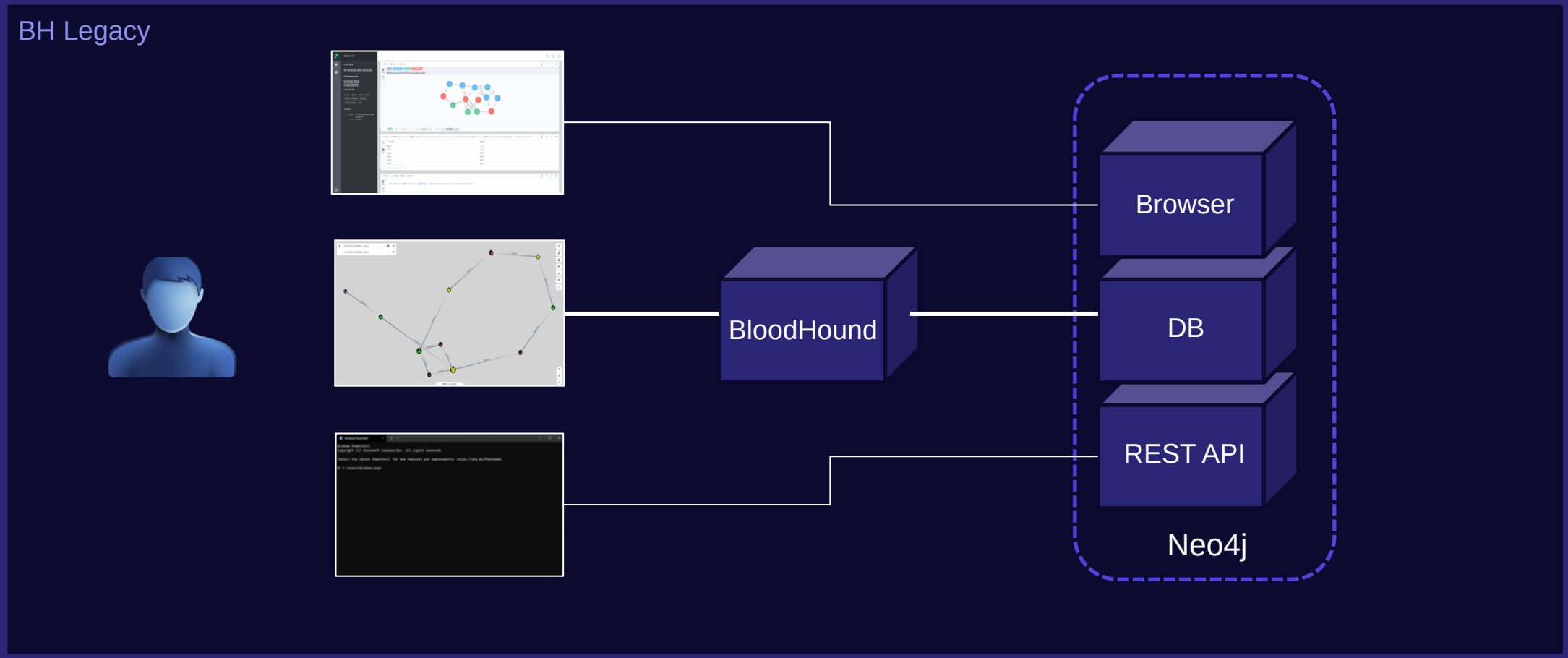
Old vs. New – Components: Old BloodHound



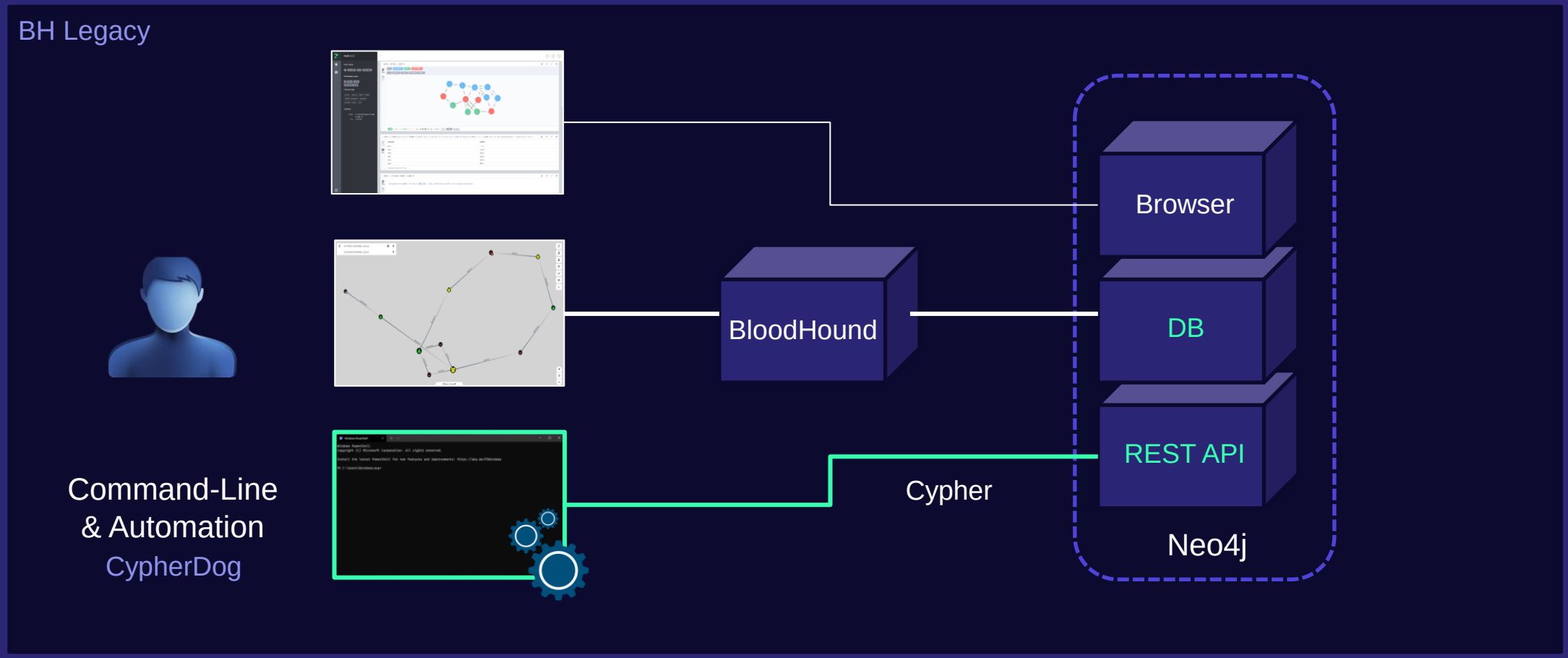
Old vs. New – Components: Old BloodHound



Old vs. New – Components: Old BloodHound

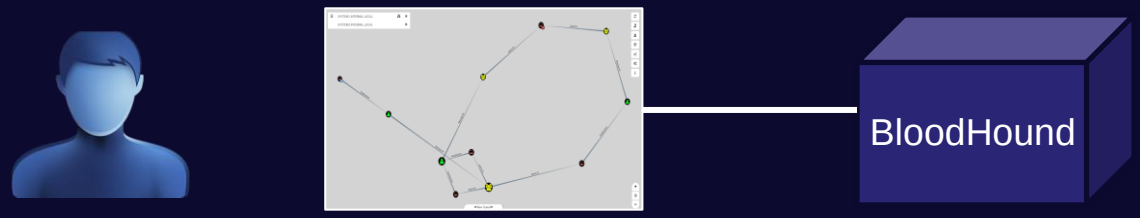


Old vs. New – Components: Old BloodHound

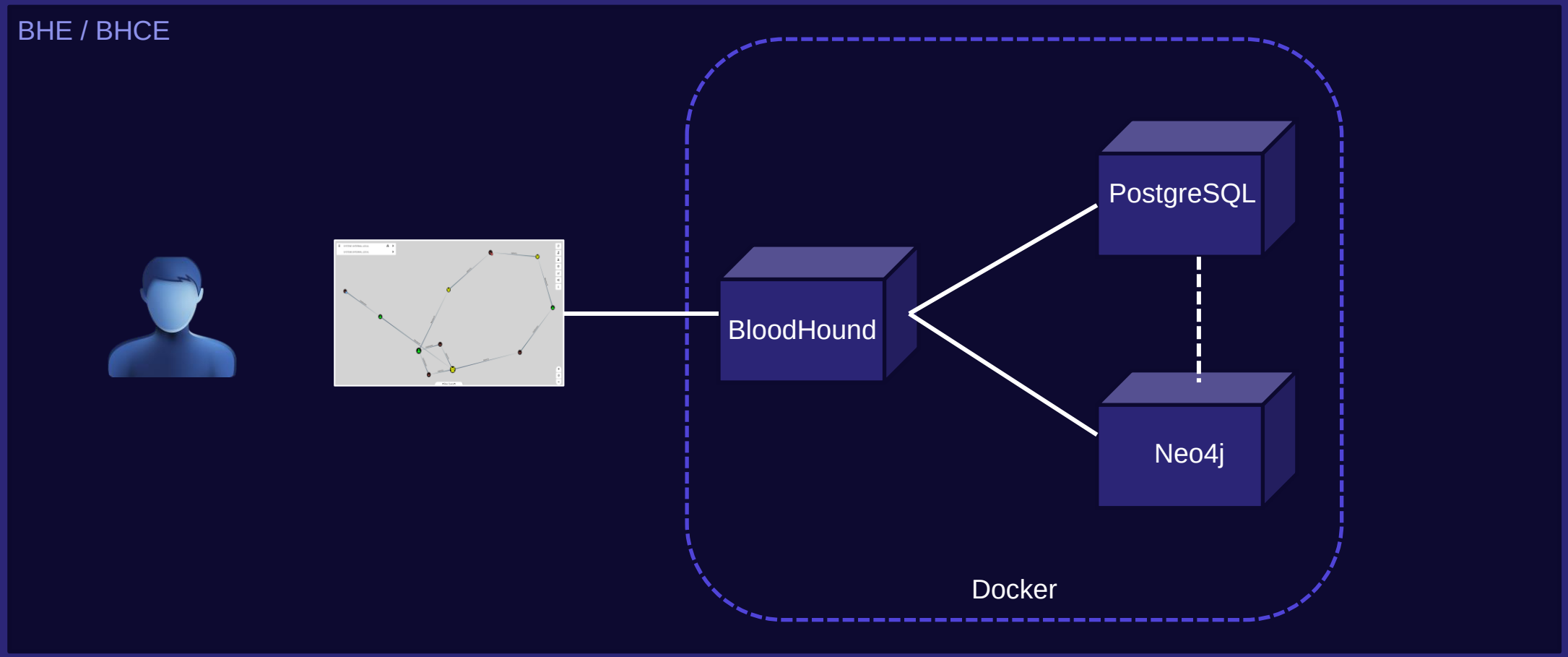


Old vs. New – Components: New BloodHound

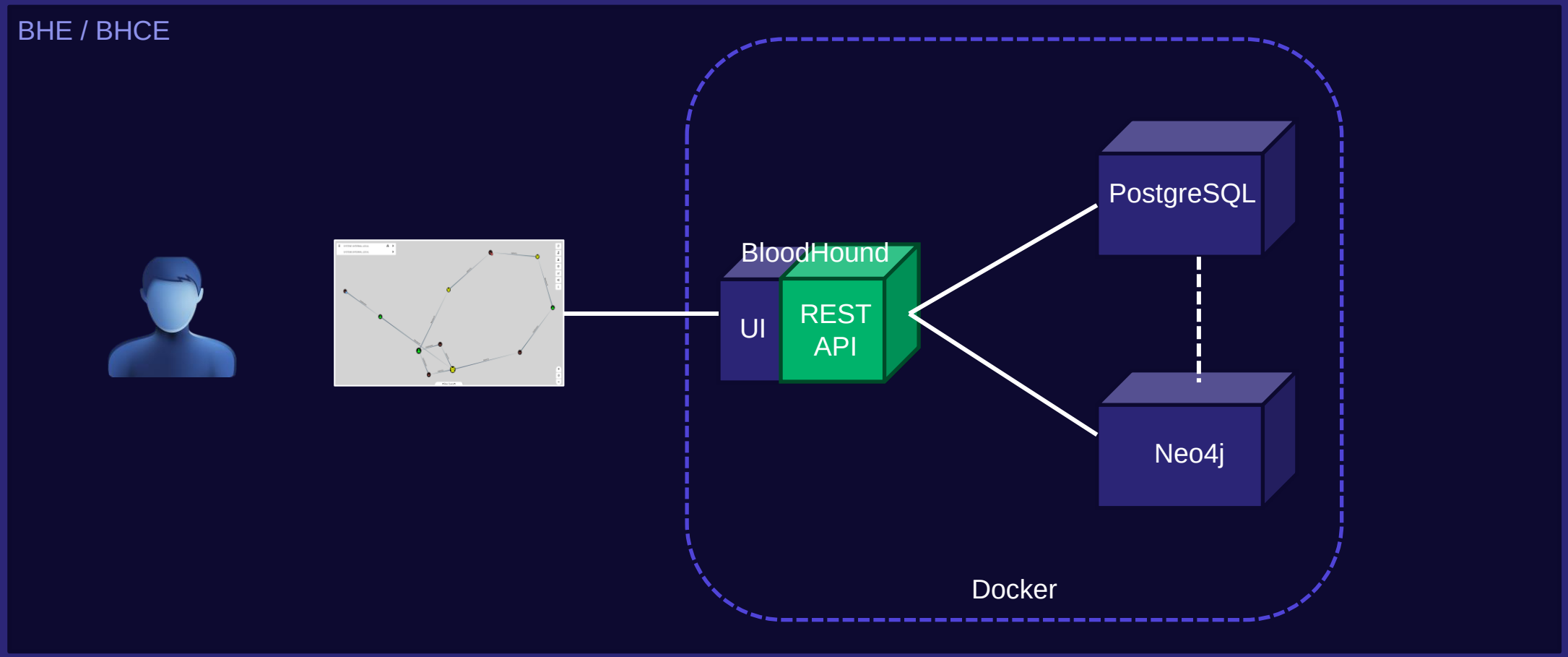
BHE / BHCE



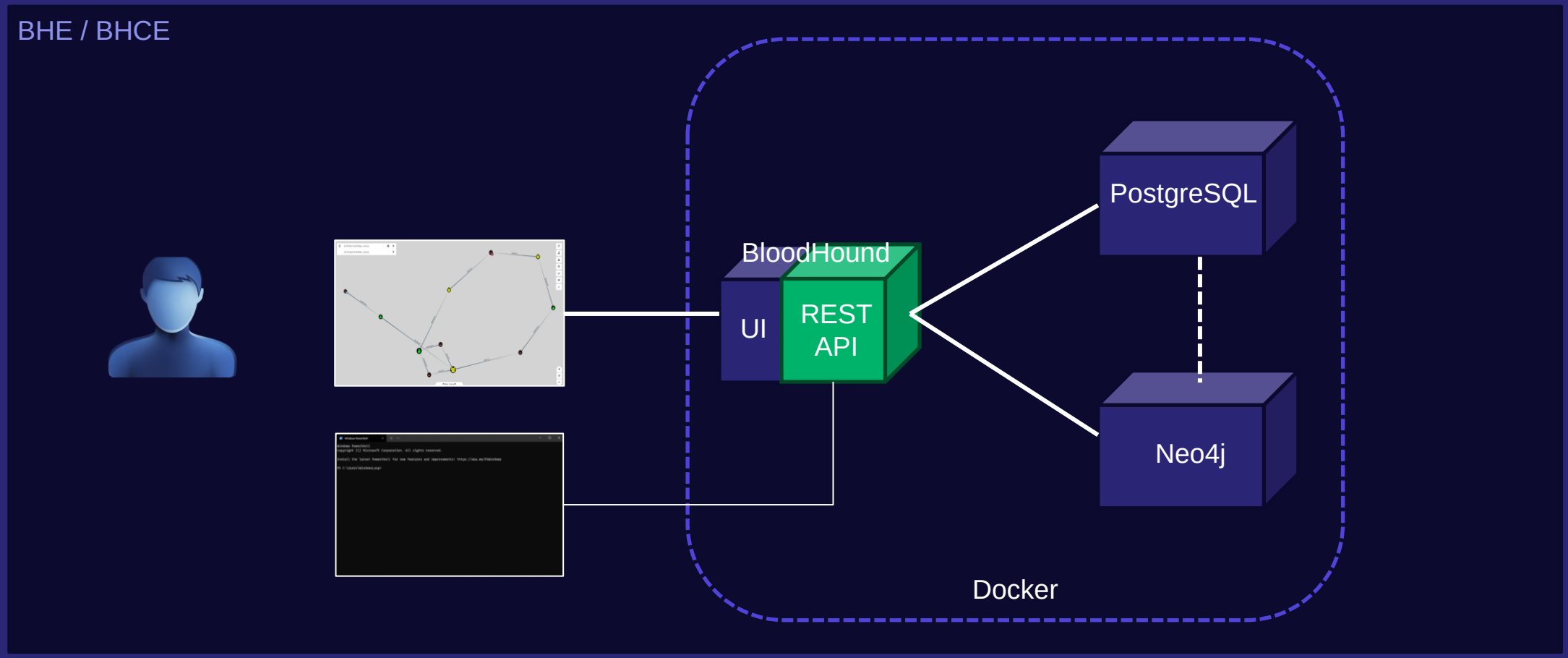
Old vs. New – Components: New BloodHound



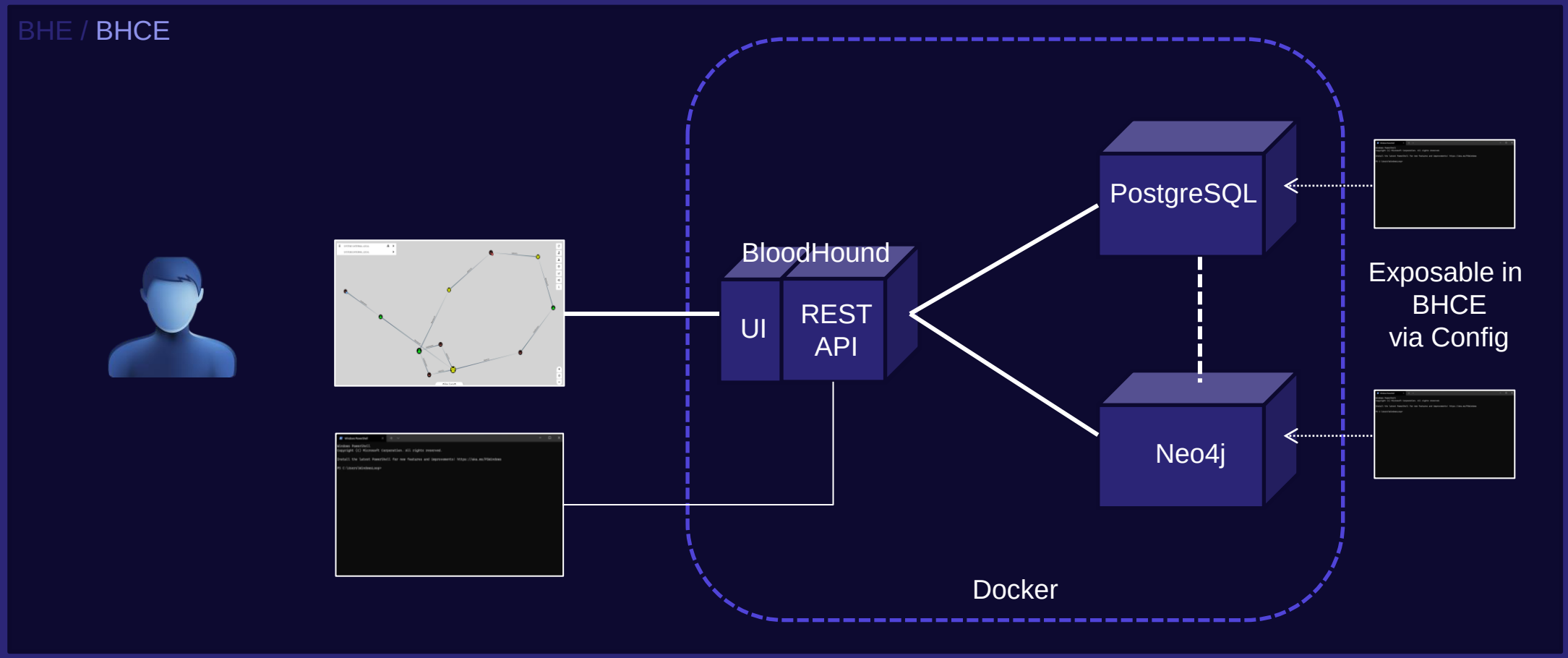
Old vs. New – Components: New BloodHound



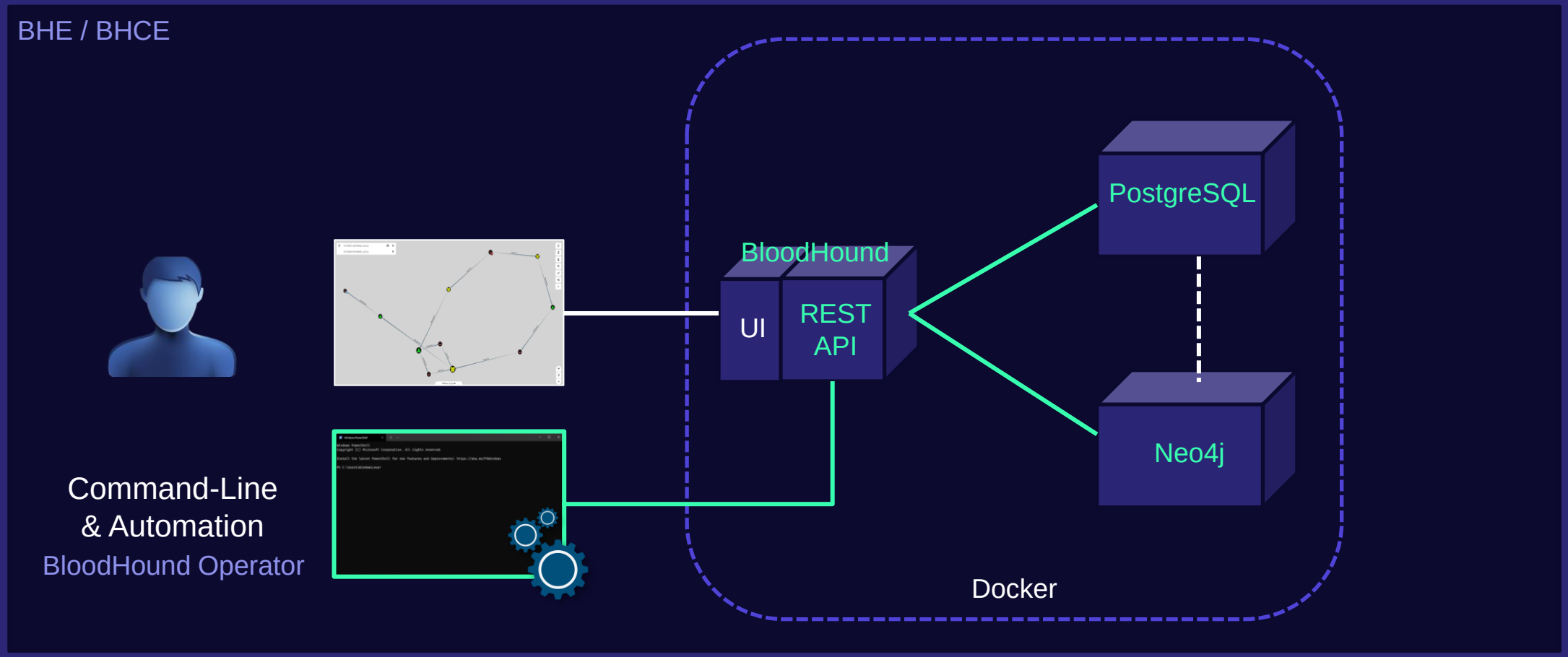
Old vs. New – Components: New BloodHound



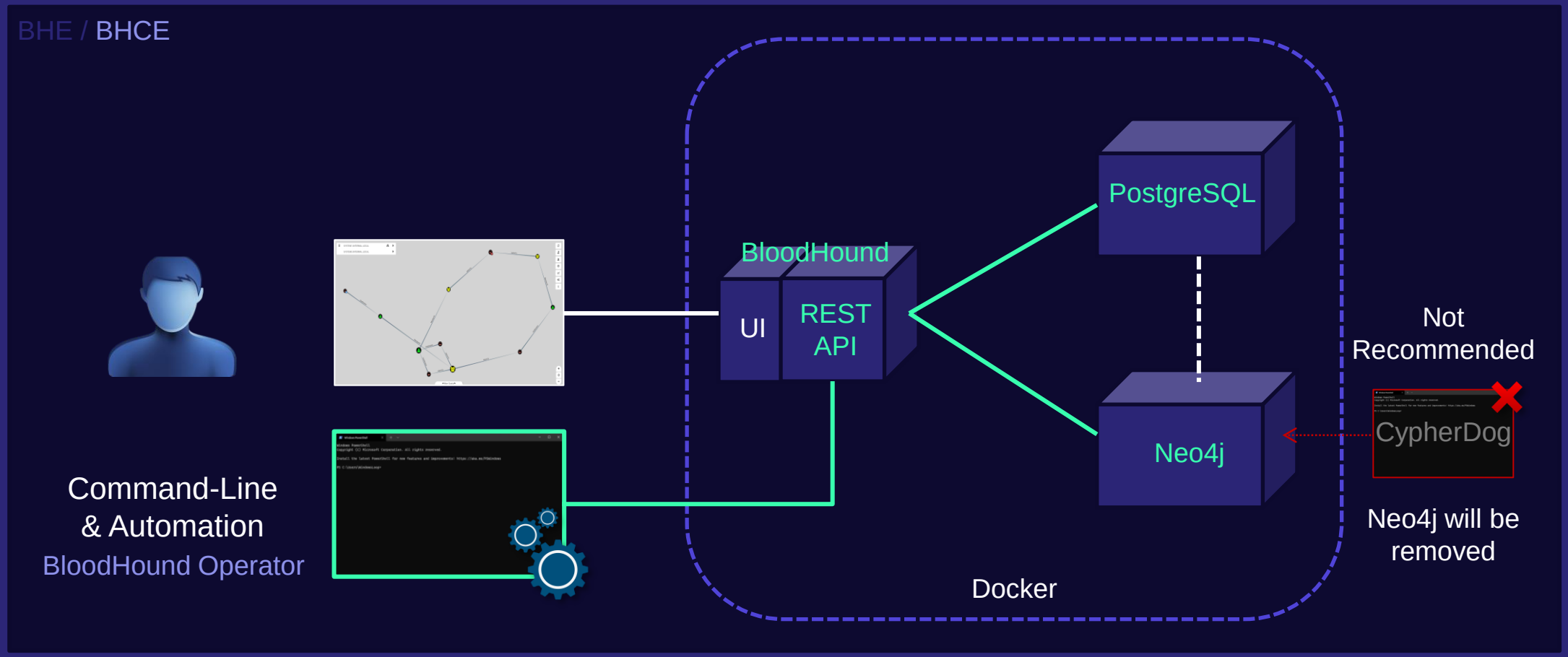
Old vs. New – Components: New BloodHound



Old vs. New – Components: New BloodHound



Old vs. New – Components: New BloodHound



Old vs. New – Feature Comparison

FEATURE	LEGACY	COMMUNITY
Easy Install	✗	✓
User Management	n/a	✓
Secure Authentication	✗	✓
Custom Queries via REST API	✓	✓ *
Data Ingestion via REST API	✗	✓
Other Actions via REST API	✗	✓

Old vs. New – Feature Comparison

FEATURE	LEGACY	COMMUNITY
Easy Install	✗	✓
User Management	n/a	✓
Secure Authentication	✗	✓
Custom Queries via REST API	✓	✓ *
Data Ingestion via REST API	✗	✓
Other Actions via REST API	✗	✓
ADCS Attack Paths	✗	✓


NO TIME TO
OLD
GET IN THE
NEW

New BloodHound REST API

BloodHound REST API – Overview

- Exposes all functionalities of UI
- Follows OpenAPI specs
- Secure Authentication
- Shared BHCE / BHE structure
- Endpoints: BHCE=142 / BHE=195
- Explorable in UI

BloodHound REST API – Roles & Permissions

Authority	Permission	Read-Only	Upload-Only	User	Power User	Admin
App	**ReadAppConfig**	✓		✓	✓	✓
App	**WriteAppConfig**				✓	✓
Auth	**CreateToken**	✓		✓	✓	✓
Auth	**ManageAppConfig**					✓
Auth	**ManageProviders**					✓
Auth	**ManageSelf**	✓		✓	✓	✓
Auth	**ManageUsers**					✓
GraphDB	**Read**	✓		✓	✓	✓
GraphDB	**Write**		✓		✓	✓
Saved_Queries	**Read**			✓	✓	✓
Saved_Queries	**Write**			✓	✓	✓
Clients	Manage				✓	✓
Clients	Read			✓	✓	✓
Clients	Taskings		✓		✓	✓
Collection	ManageJobs				✓	✓
Risks	GenerateReport	✓		✓	✓	✓
Risks	ManageRisks				✓	✓

Access to API endpoints is determined by role and associated permissions in the application

BHE only

BloodHound REST API – Authentication

- Uses TokenID + TokenKey (incl. User Token Lifecycle)
- Each request must be signed
- 3x HMAC SHA256
- Signature = [Hash:(((TokenKey x Operation) x Timestamp) x Body)].ToBase64
- Headers must include TokenID, Signature and Timestamp
- Server validates signature against request

Working with the BloodHound API – BloodHound (bloodhoundenterprise.io)

BloodHound REST API – Authentication

PowerShell

```powershell
# Signature
$Timestamp = [Datetime]::utcnow.tostring('o')
$KeyByte   = [Text.Encoding]::UTF8.GetBytes($TokenKey)
$OpByte    = [Text.Encoding]::UTF8.GetBytes("$Method/$URI")
$DateByte  = [Text.Encoding]::UTF8.GetBytes(-join $Timestamp[0..12])
$BodyByte  = [Text.Encoding]::UTF8.GetBytes("$Body")
$HMAC      = [Security.Cryptography.HMACSHA256]::new($KeyByte).ComputeHash($OpByte)
$HMAC      = [Security.Cryptography.HMACSHA256]::new($HMAC).ComputeHash($DateByte)
$HMAC      = [Security.Cryptography.HMACSHA256]::new($HMAC).ComputeHash($BodyByte)
$Sign      = [Convert]::ToBase64String($HMAC)
```

```powershell
# Headers
$Headers = @{
    Authorization = "BHESignature $TokenID"
    Signature     = $Sign
    RequestDate   = $Timestamp
    }
```

PowerShell BloodHound Operator

BloodHound Operator – Features

- PowerShell Cmdlets for the BloodHound REST API (50+)
  - Verb-BHNoun Syntax / Pipeline Input / Output Objects / Tab-Completion / …
  - All application features are accessible
  - Allows for scripted deployments
  - Allows for full Ingest-Query-Analyze automation (DIY)
  - Experimental: Multiple sessions

BloodHound Operator – Disclaimer

- Still a Work-In-Progress…
- ADCS / BHE endpoints not included yet
- Some things will change = Break
- Provided as Gist for beta testers / No Support (feedback via BH Slack)
- Not an official SpecterOps tool
- Use at your own risks…

¯_(ツ)_/¯

DEMO

BloodHound Operator – Basic Usage

What’s Next ?

ToDo

ToDo

- On my side:
- Add ADCS / BugFix / Docs
- Add BHE functionalities
- Format as proper Module
- Publish to PS Gallery
- Present at PSConfEU

ToDo

- On my side:
- Add ADCS / BugFix / Docs
- Add BHE functionalities
- Format as proper Module
- Publish to PS Gallery
- Present at PSConfEU

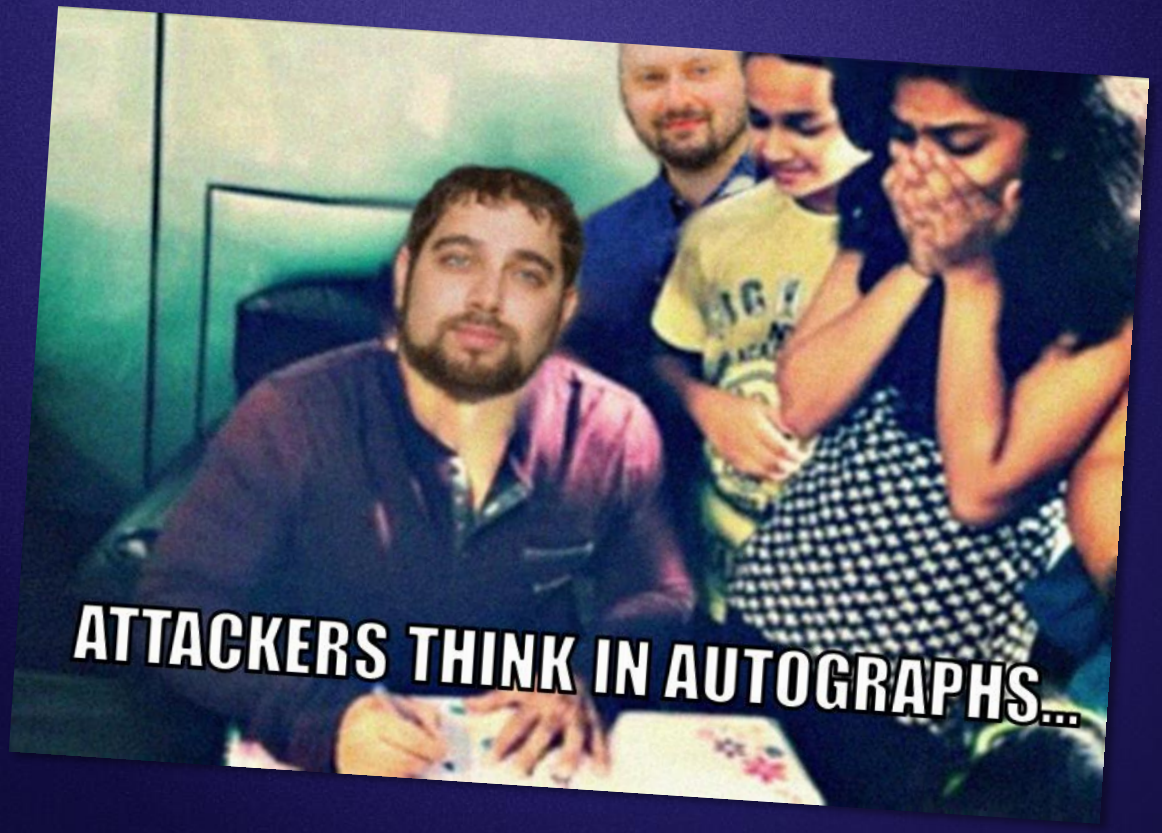
- On your side:
- Download Gist
- Test-drive
- Feedback via BH Slack
- Wait for release
- Automate all the things…

Questions ?

Reminder

CptJesus will be signing autographs in the lobby later this afternoon…

Come and get yours!



SPECTEROPS

Thank you

Catch me in the lobby if you want to chat…

Enjoy the rest of SO-CON!

SadProcessor

SPECTEROPS