BlokScript

A programming language for Storyblok.

https://www.blokscript.com/

Problem

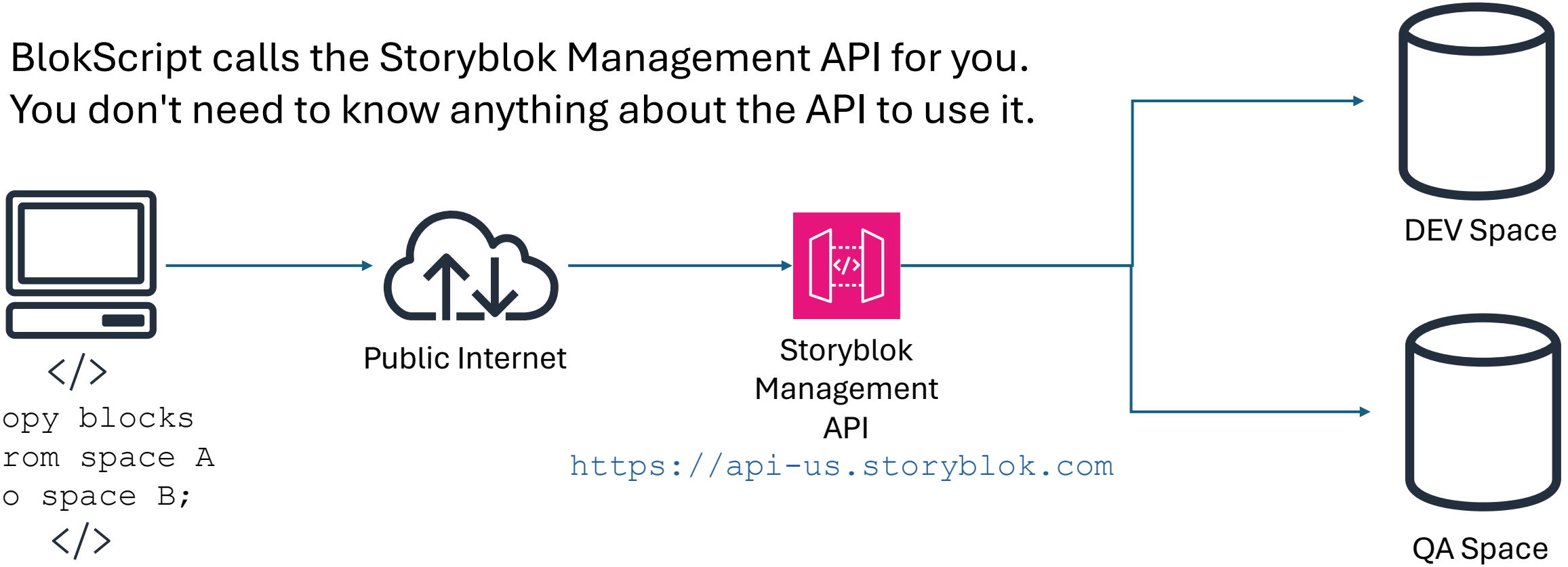
```
copy whatever we want in storyblok
from space A
to space B
where level of effort = low.
```

Solution

```
//
// ONE BLOKSCRIPT STATEMENT SOLVES THIS PROBLEM.
//
copy blocks
     from space 'Advance'
     to space 'Napa'
where name matches regex '^oil'
     or name ends with 'viscosity';
```

How It Works

BlokScript calls the Storyblok Management API for you.
You don't need to know anything about the API to use it.




```
</>
copy blocks
from space A
to space B;
</>
```

Language & Interpreter Design Goals

- **Domain specific**. Storyblok native concepts are built-in.
- **Solution driven**. Primary goal is to move things around.
- **Explicit, easy to read.** At the cost of being “noisy” to write.
- **SQL-like**. Similar operations and constraints have a natural fit.
- **Larger user audience**. Users that don't know APIs can use it.
- **Informative**. BlokScript gives you stats & tells you what it's doing.

Copying Blocks

```
//
// COPY BLOCKS TO ANOTHER SPACE.
// THE BLOCK IS CREATED OR UPDATED.
// SEE ALSO: delete blocks
//
copy blocks
     from space 'Advance'
     to space 'Napa'
where name like 'oil%';
```

Copying Stories

```
//
// COPY SPECIFIC STORIES TO ANOTHER SPACE.
// SEE ALSO: delete, publish, unpublish.
//
copy stories
     from space 'Advance'
     to space 'Napa'
where url starts with '/penn';
```

Publishing Stories

```
//
// PUBLISH SELECTED STORIES IN A SPACE.
// SEE ALSO: unpublish stories
//
publish stories in space 'Napa'
where any tag in ('product', '2024-07-15');
```

Managing Datasources

```
//
// CREATE A DATASOURCE. COPY TO ANOTHER SPACE.
// SEE ALSO: update, delete,
//
create datasource 'Competitors' in space 'Advance';
copy datasources in space 'Advance' to space 'Napa';
```

Managing Datasource Entries

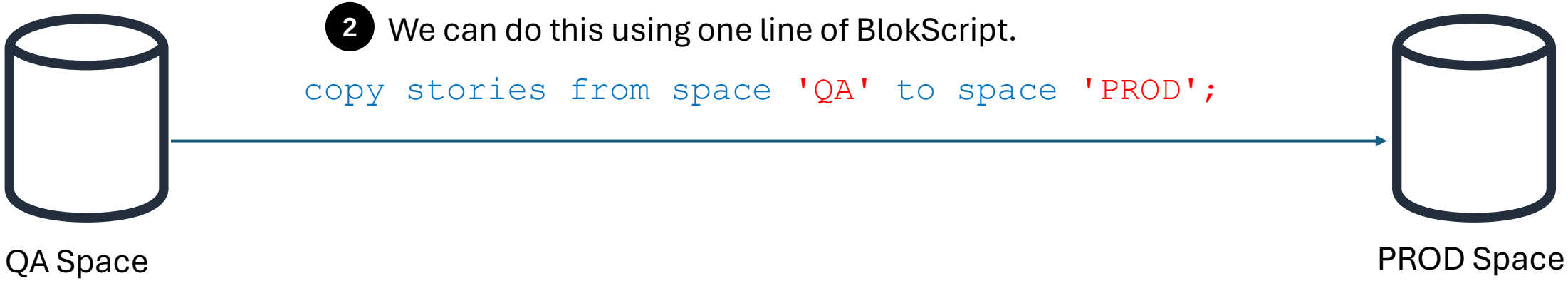
```
//
// CREATE A DATASOURCE ENTRY.
// SEE ALSO: update, delete, copy
//
create datasource entry 'AutoZone'
     in datasource 'Competitors'
     in space 'Advance';
```

Variables

```
//
// int, string, regex, block, datasource, story
//
var Advance = space 'Advance';
var Napa = space 'Napa';

copy stories from Advance to Napa
where name starts with 'Schaeffer';
```

Content Promotion



1

A large amount of content is created in a pre-production environment. Instead of manually recreating that content, we want to “promote” it to PROD.

Slicing

1

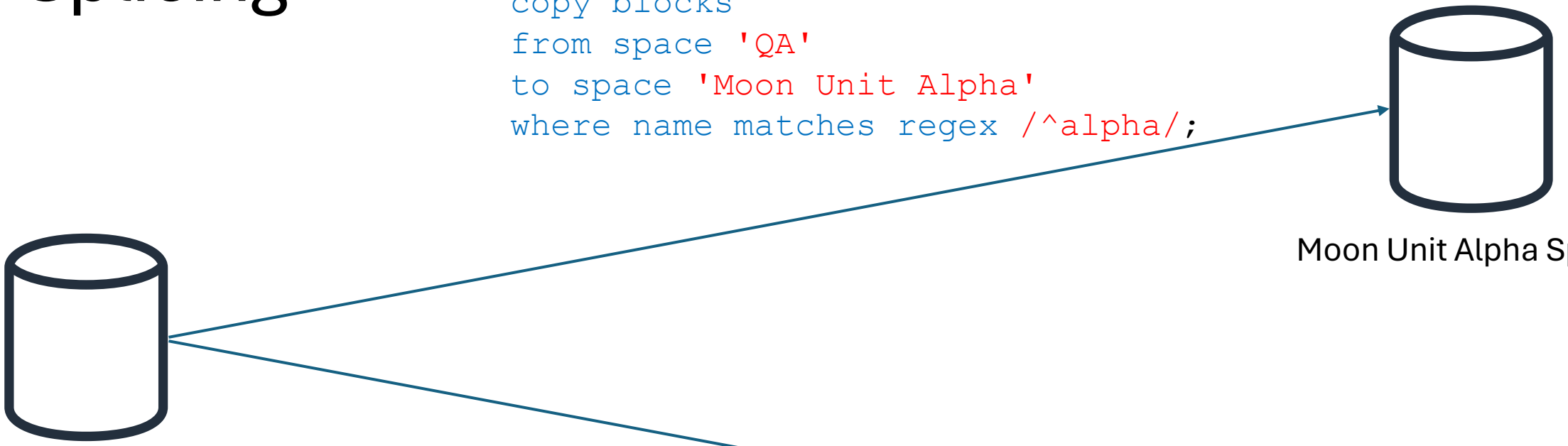
Spaces are expensive, so the DEV and QA schemas both support multiple sites. We want to promote only the Napa blocks to QA.

2

We can slice by content and schema. The intent here is to improve development operations while using less spaces.

Splicing

```
copy blocks
from space 'QA'
to space 'Moon Unit Alpha'
where name matches regex /^alpha/;
```



Moon Unit Alpha Space

QA Space

```
copy blocks
from space 'QA'
to space 'Moon Unit Zappa'
where name matches regex /^zappa/;
```

Moon Unit Zappa Space

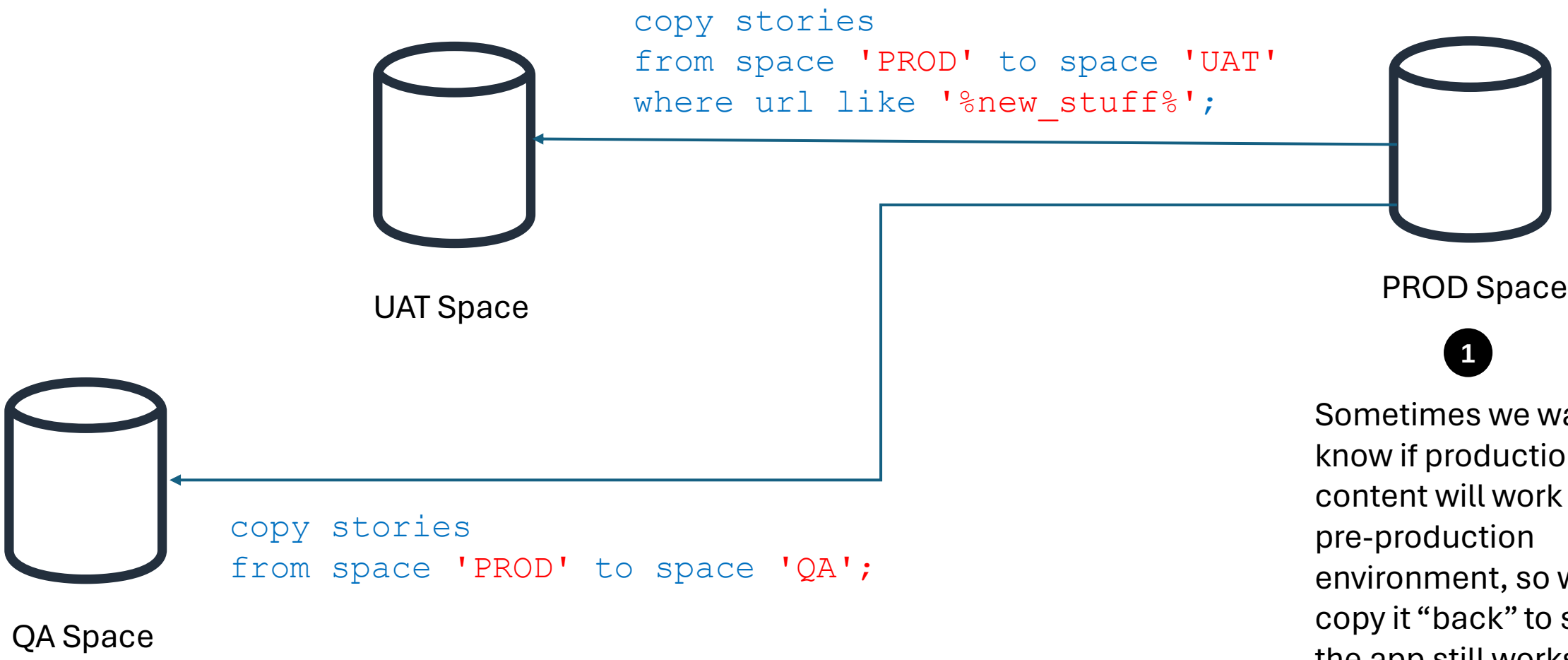
1

Spaces are expensive, so the QA schema supports multiple sites. We want to move those blocks to the appropriate space in PROD.

2

We can splice by content and schema. The intent here is to take advantage of the performance and reliability of multiple spaces.

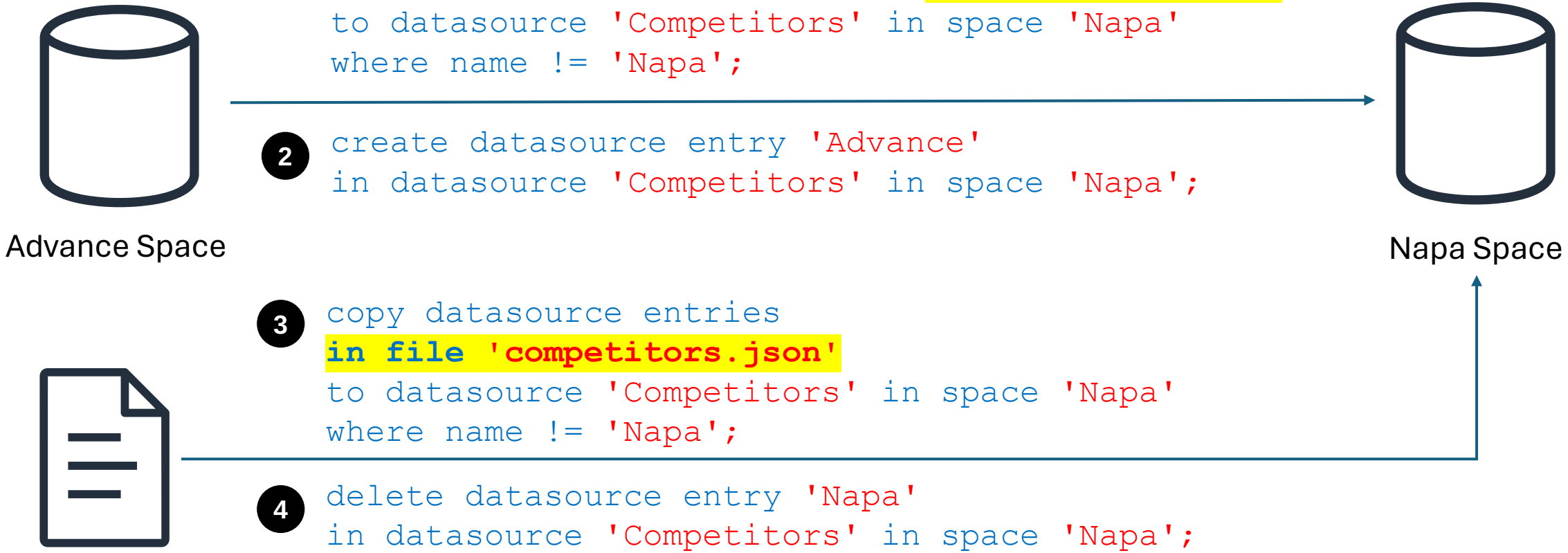
“Backcopying” Content For Testing



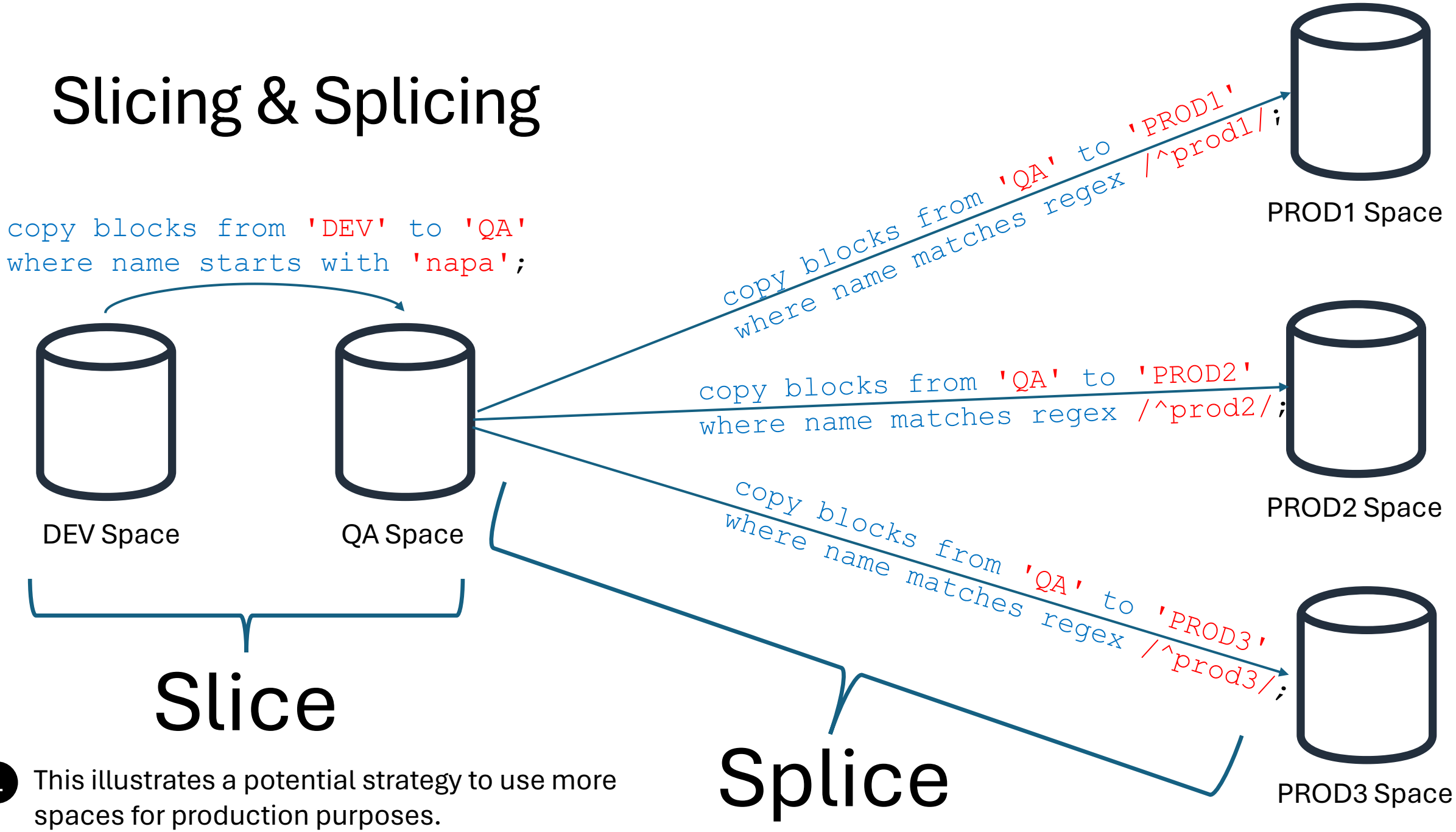
1

Sometimes we want to know if production content will work in a pre-production environment, so we copy it “back” to see if the app still works.

Datasource Loading



Slicing & Splicing



1. This illustrates a potential strategy to use more spaces for production purposes.

Other BlokScript Features

- Automatic local caching of space objects to reduce API calls.
- Automatic throttling, recovery & retry of API calls.
- Automatic selects the right create or update API method and/or endpoint to call.
- Syntactic sugar to make the code writing experience better. These space literals are all equivalent:
  - `space 1234567`
  - `space '1234567'`
  - `space '#1234567'`
- You can also save to files.
  - `copy spaces to file 'spaces.json';`

BlokScript Implementation Details

- Written in C# and compatible with every .NET runtime.
- ANTLR for lexer and parser generation.
- Newtonsoft for JSON processing.

BlokScript Is Free & Open Source (GNU v3)

- Code and website both open sourced to the community.
- https://github.com/cwses1/blokscript
- https://github.com/cwses1/blokscript-dot-com

End of Presentation Thoughts

- BlokScript (BS) Jokes
  - “Just *.bs your way through it like you do every other problem.”
  - “A little bit of BS goes a long way.”
- Deep Thoughts
  - “Why Program by Hand in Five Days what You Can Spend Five Years of Your Life Automating?” - Terence Parr
  - “Civilization advances by extending the number of important operations which we can perform without thinking of them. ” - Alfred North Whitehead