Cyber Security - An Engineer's Perspective

Tejas Parnerkar

Software development engineer

Prospa

Tejas Parnerkar

I'm an experienced software engineer with experience developing solutions across a range of industries, which include Finance, Banking, Media, Logistics and Primary Sector.

I specialize in full stack application development, using languages and frameworks such as C#/ASP.Net, Java, Angular and React.

I enjoy using technology to come up with innovative solutions to business use cases.

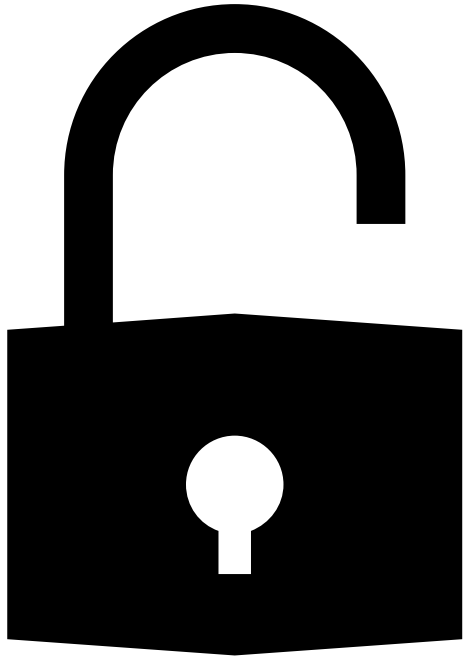
I hold a degree in Computer Systems Engineering from the University of Auckland.

Agenda

Broken Access Control

Zero Trust

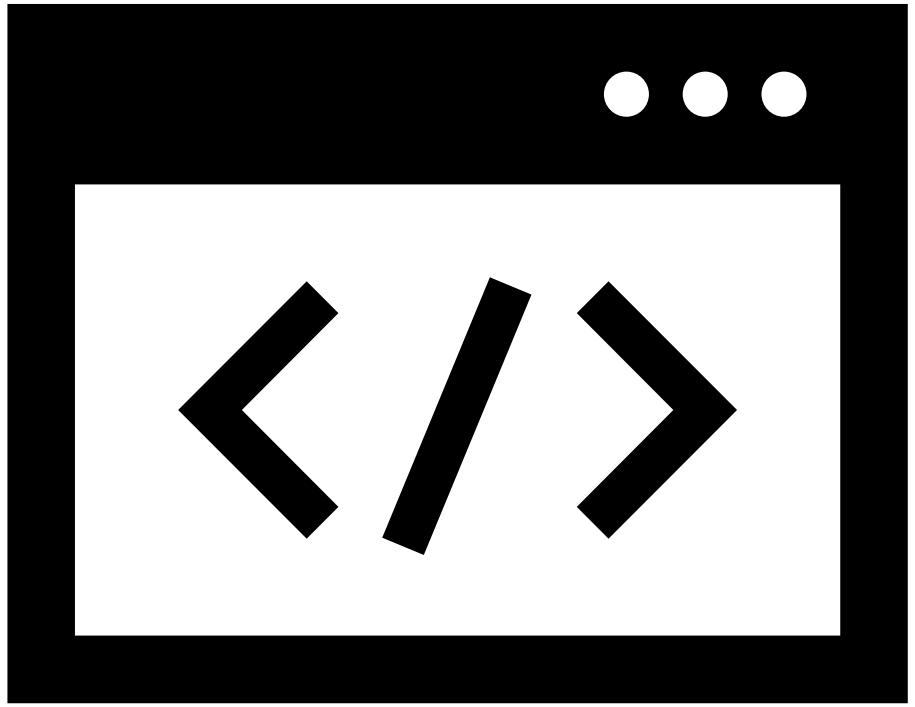
Secret Management



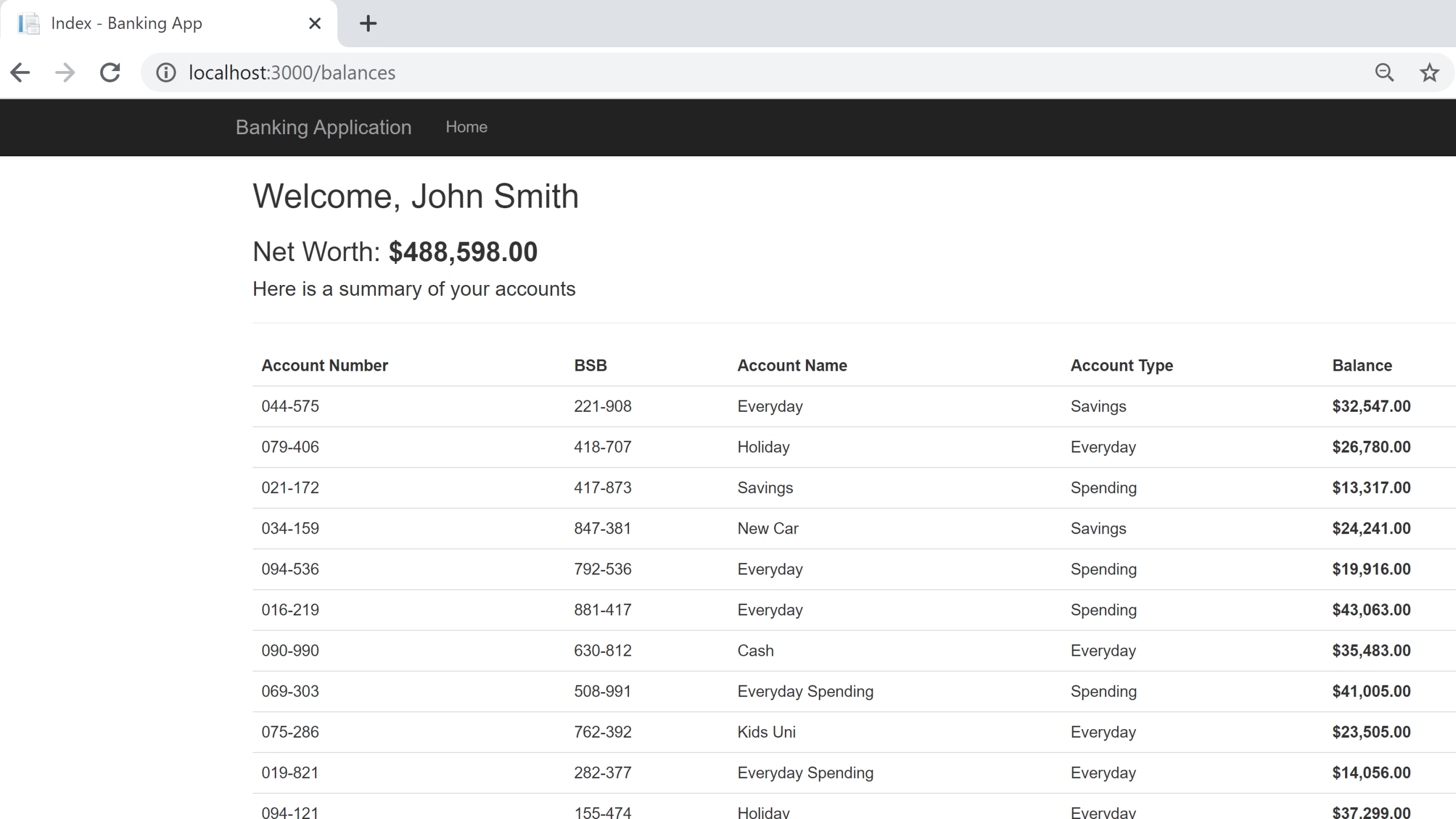
Broken Access Control

Broken Access Control

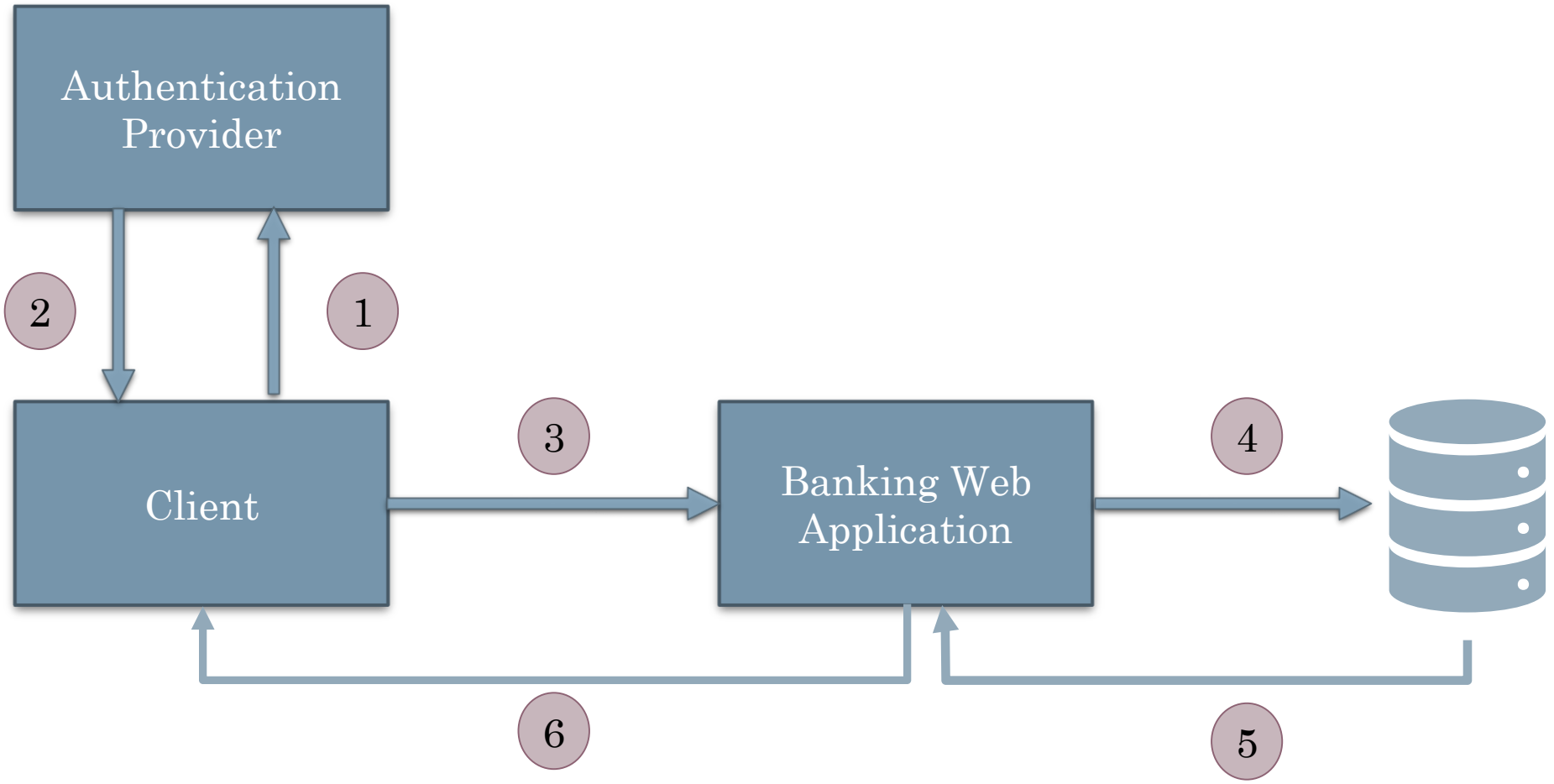
- Bypassing checks
  - Modifying URLs, Application state or API calls, illegally to violate intended system functions
- Primary Key
  - If database primary keys are exposed, an attacker may be able to access someone else's records
- Elevation of Privilege
  - E.g. logged in user can perform admin functions



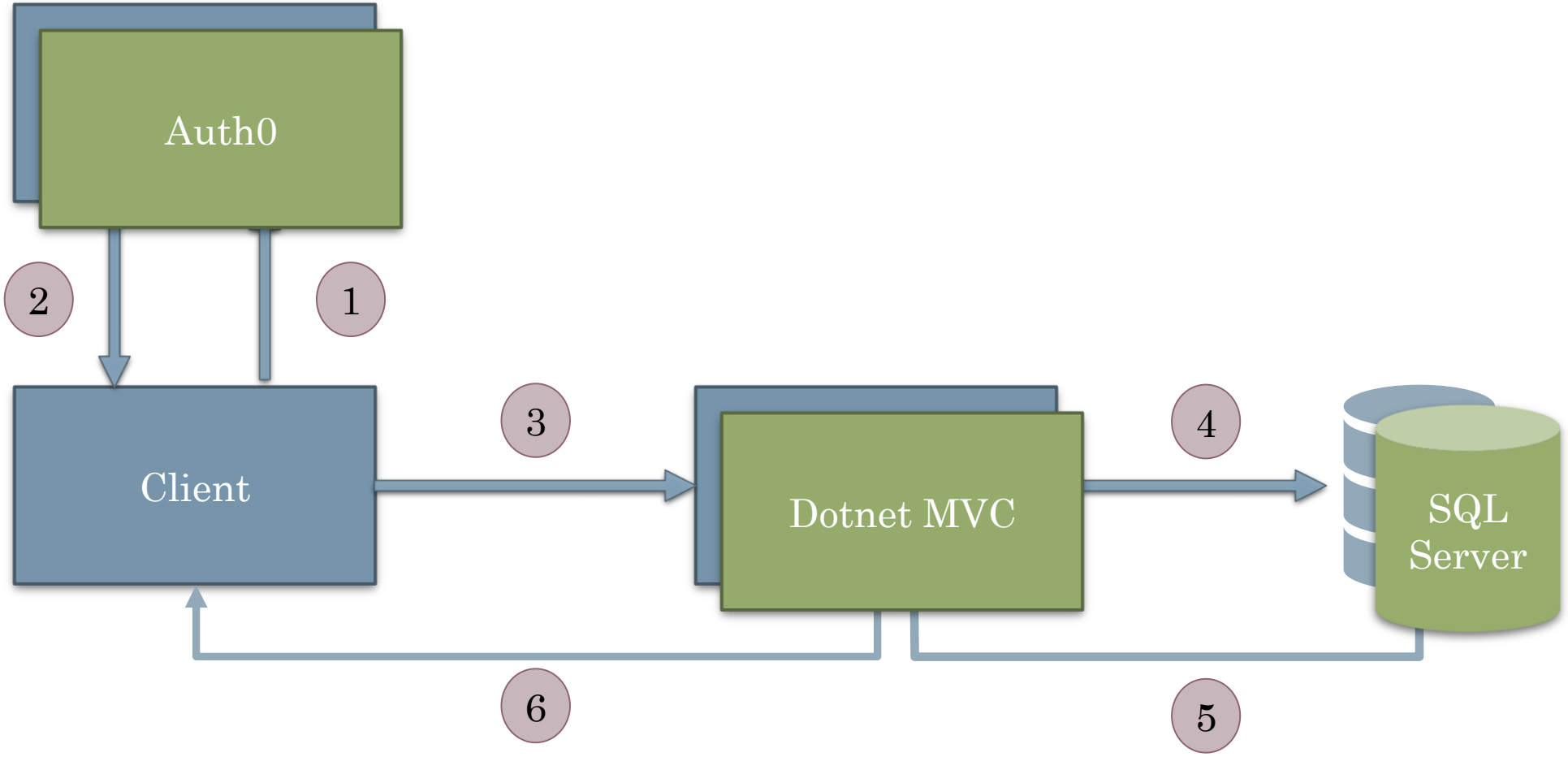
Example Application


Index - Banking App
localhost:3000/balances
Banking Application
Home
Welcome, John Smith
Net Worth: $488,598.00
Here is a summary of your accounts
Account Number
BSB
Account Name
Account Type
Balance
044-575
221-908
Everyday
Savings
$32,547.00
079-406
418-707
Holiday
Everyday
$26,780.00
021-172
417-873
Savings
Spending
$13,317.00
034-159
847-381
New Car
Savings
$24,241.00
094-536
792-536
Everyday
Spending
$19,916.00
016-219
881-417
Everyday
Spending
$43,063.00
090-990
630-812
Cash
Everyday
$35,483.00
069-303
508-991
Everyday Spending
Spending
$41,005.00
075-286
762-392
Kids Uni
Everyday
$23,505.00
019-821
282-377
Everyday Spending
Everyday
$14,056.00
094-121
155-474
Holiday
Everyday
$37,299.00

A Sample Application



A Sample Application



A Demo

Some Conclusions

Do not pass sensitive info in url's

Do not show developer error messages in Production Environments

When retrieving a database record, ensure the intended audience ONLY access it

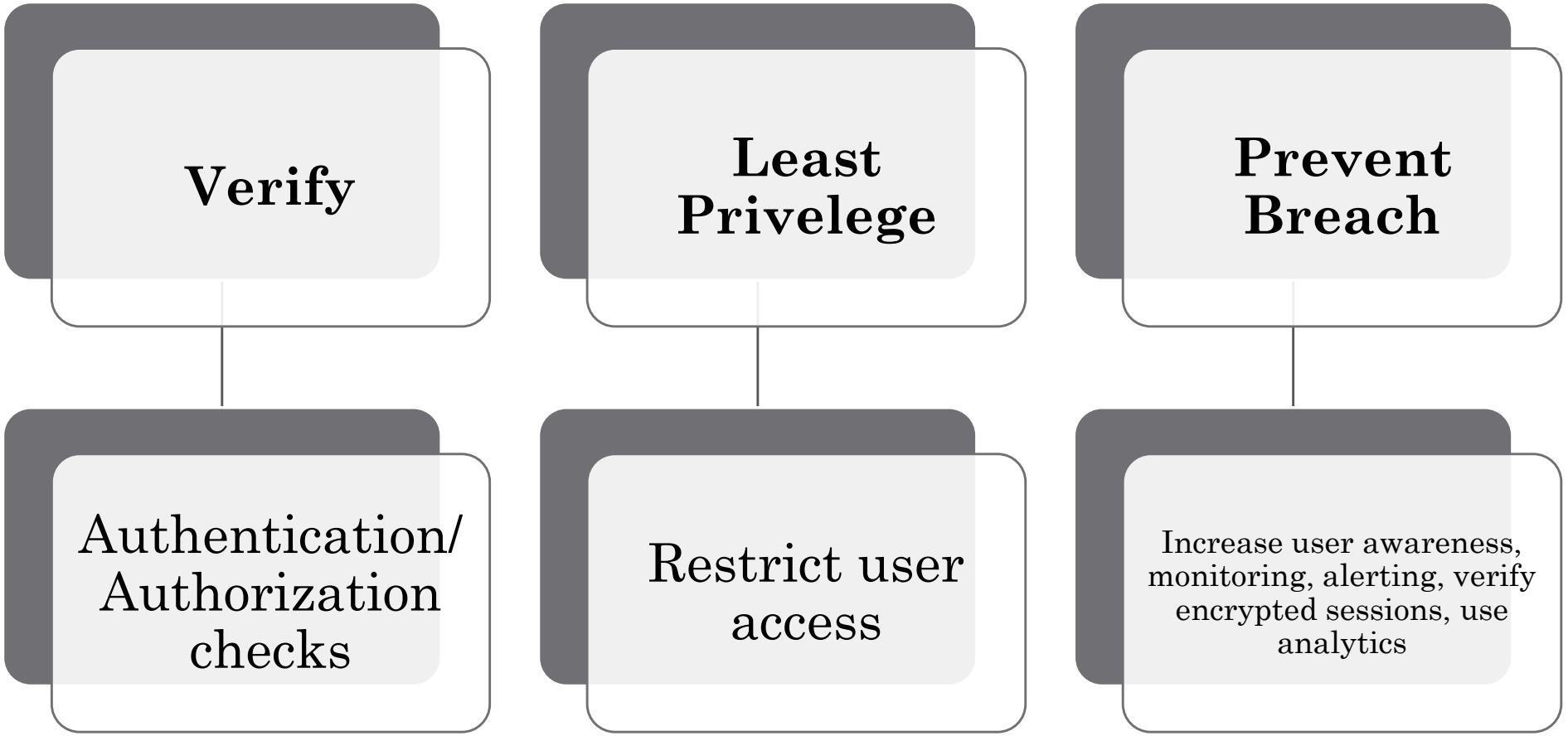
Some Conclusions - URLs

https://banking-application.com/id/5 - BAD

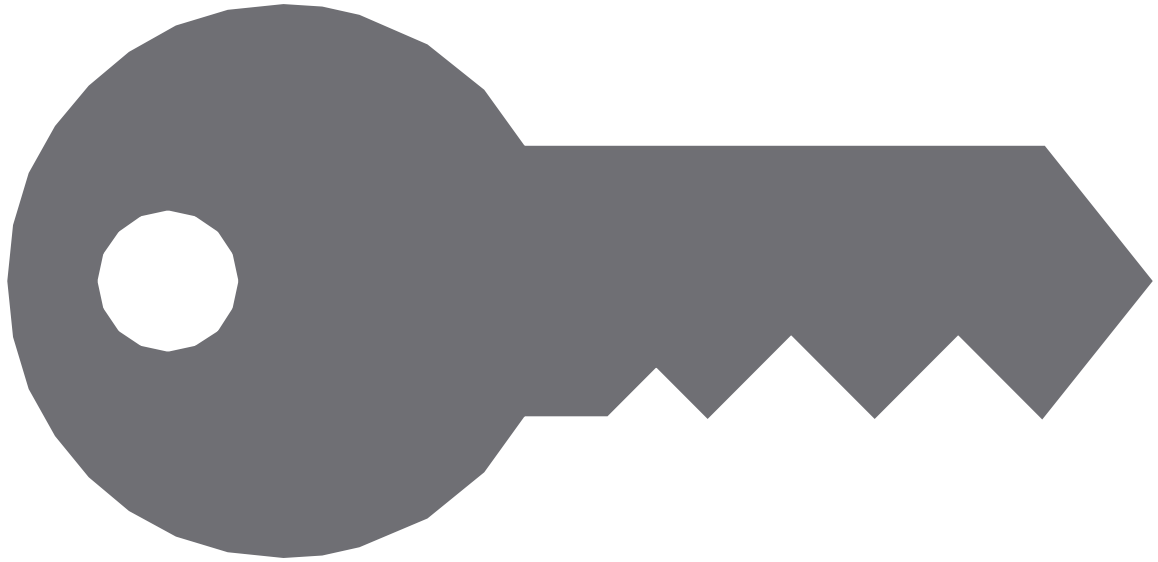
https://banking-application.com/home?firstName=foo&lastName=bar - BAD

https://banking-application.com/ - GOOD

Zero Trust Principles



Source: https://www.microsoft.com/en-au/security/business/zero-trust



Secret Management

appsettings.json

Schema: https://json.schemastore.org/appsettings

```
{
  "Logging": {
    "IncludeScopes": false,
    "LogLevel": {
      "Default": "Debug",
      "System": "Information",
      "Microsoft": "Information"
    }
  },
  "kestrel": {
    "EndPoints": {
      "Http": {
        "Url": "http://*:3000"
      }
    }
  },
  "Auth0": {
    "Domain": "",
    "ClientId": "",
    "ClientSecret": ""
  },
  "ConnectionStrings": {
    "db": ""
  }
}
```

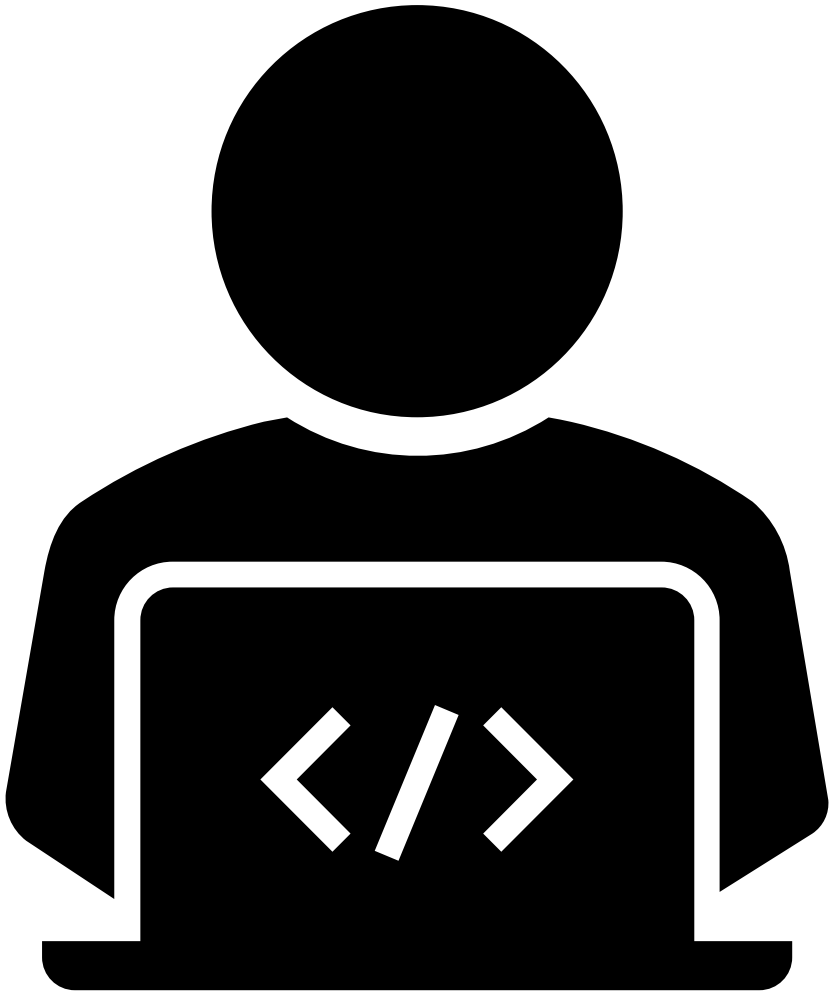
Secret Management

- Connection strings, API Keys, Secrets SHOULD NOT site in source code
- Azure KeyVault, AWS Secrets Manager
- Inject secrets in config files on CI/CD pipelines or environment variables in containers
- Rotate on a regular basis

“It takes 20 years to build a reputation and few minutes of cyber-incident to ruin it.” –

Stephane Nappo

CISO - Société Générale



Thank You