PROJETO DE SIRS 23/24

BOMB APPETIT

RESTAURANT APP SECURE AND RELIABLE COMMUNICATION

REALIZED BY:

LUÍS MARQUES Nº 99265

RAMIRO MOLDES Nº 99313

OBJECTIVES

By the end of this presentation , students will be able to:

1

How The Document was Secured

2

How the Infrastructure was designed

3

How was configured the security of the connections and key distribuition

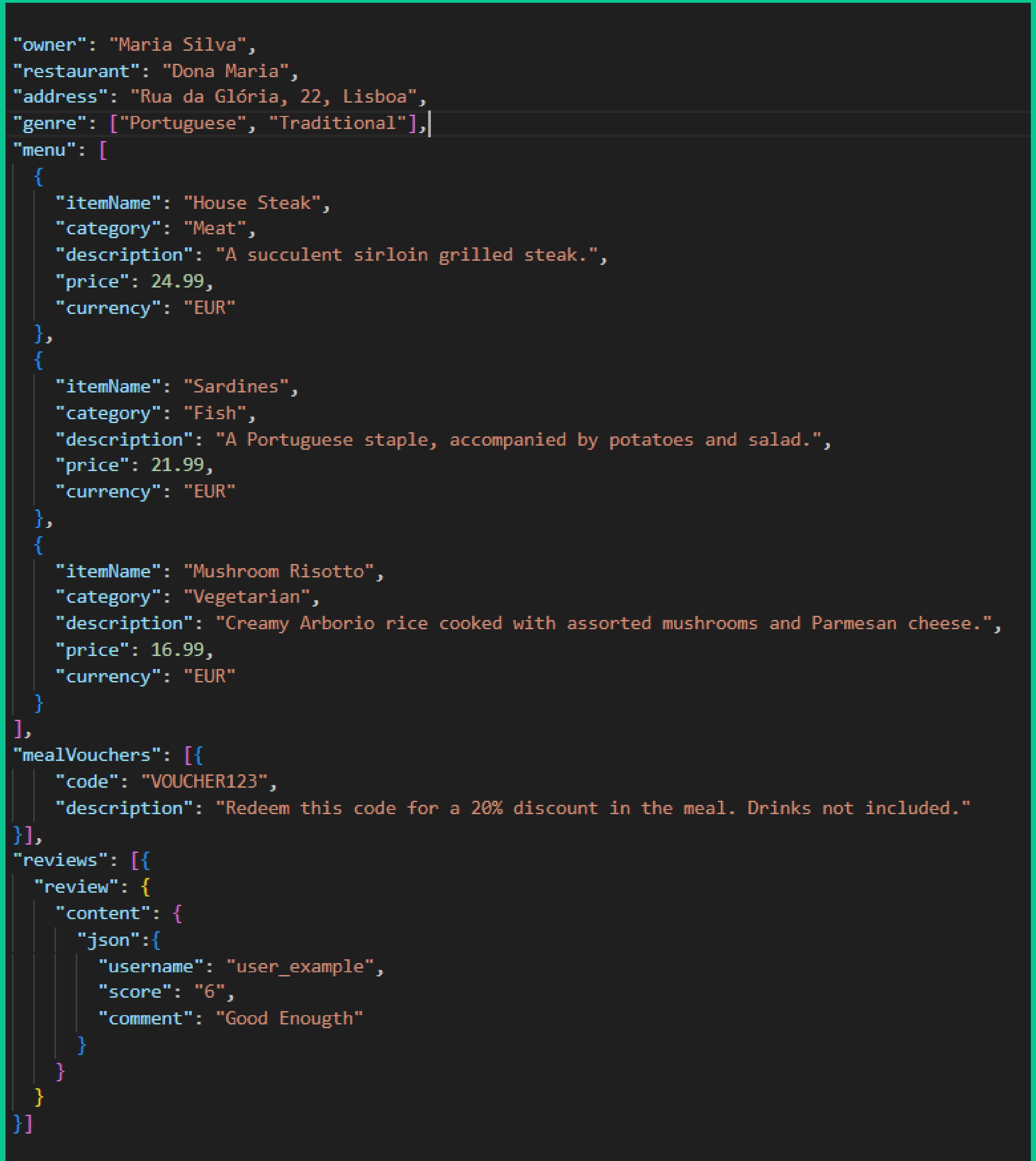
4

How it was hadled the security challenge

5

Conclusion, mishaps and realizations of what was done

DOCUMENT EXAMPLE



```
"owner": "Maria Silva",
"restaurant": "Dona Maria",
"address": "Rua da Glória, 22, Lisboa",
"genre": ["Portuguese", "Traditional"],
"menu": [
  {
    "itemName": "House Steak",
    "category": "Meat",
    "description": "A succulent sirloin grilled steak.",
    "price": 24.99,
    "currency": "EUR"
  },
  {
    "itemName": "Sardines",
    "category": "Fish",
    "description": "A Portuguese staple, accompanied by potatoes and salad.",
    "price": 21.99,
    "currency": "EUR"
  },
  {
    "itemName": "Mushroom Risotto",
    "category": "Vegetarian",
    "description": "Creamy Arborio rice cooked with assorted mushrooms and Parmesan cheese.",
    "price": 16.99,
    "currency": "EUR"
  }
],
"mealVouchers": [{
    "code": "VOUCHER123",
    "description": "Redeem this code for a 20% discount in the meal. Drinks not included."
}],
"reviews": [{
  "review": {
    "content": {
      "json":{
        "username": "user_example",
        "score": "6",
        "comment": "Good Enougth"
      }
    }
  }
}]
```

SECURITY IMPLEMENTED

Confidentiality

- AES-256 sym key
- CBC Mode
- IV
- Base 64 encoding

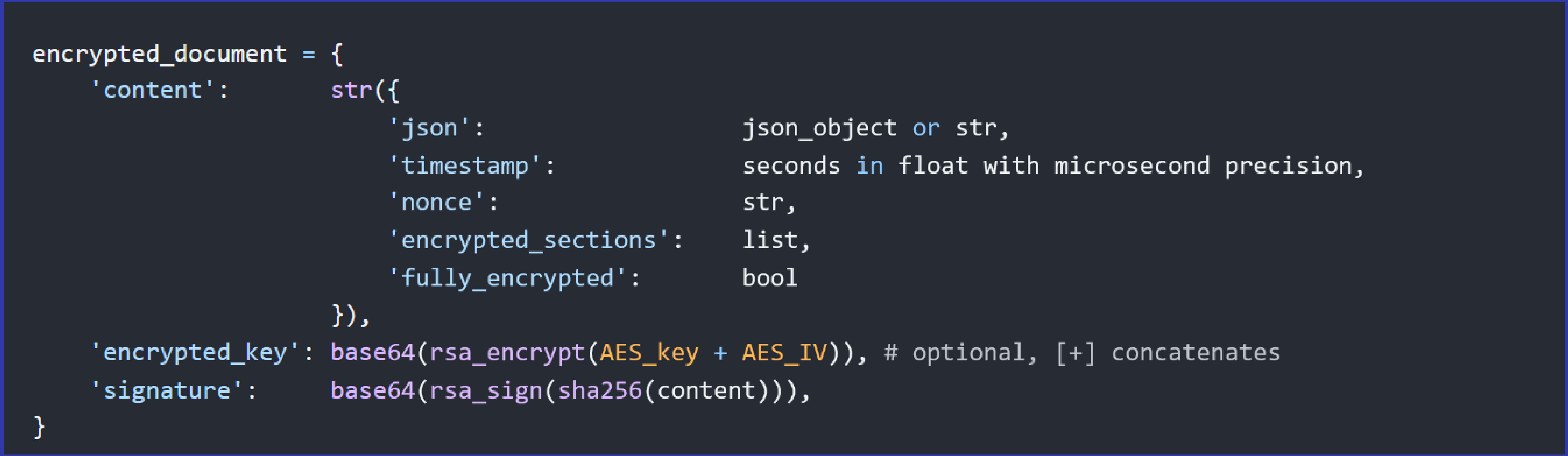
Freshness

- Timestamp
- Nonce

Authenticity and Integrity

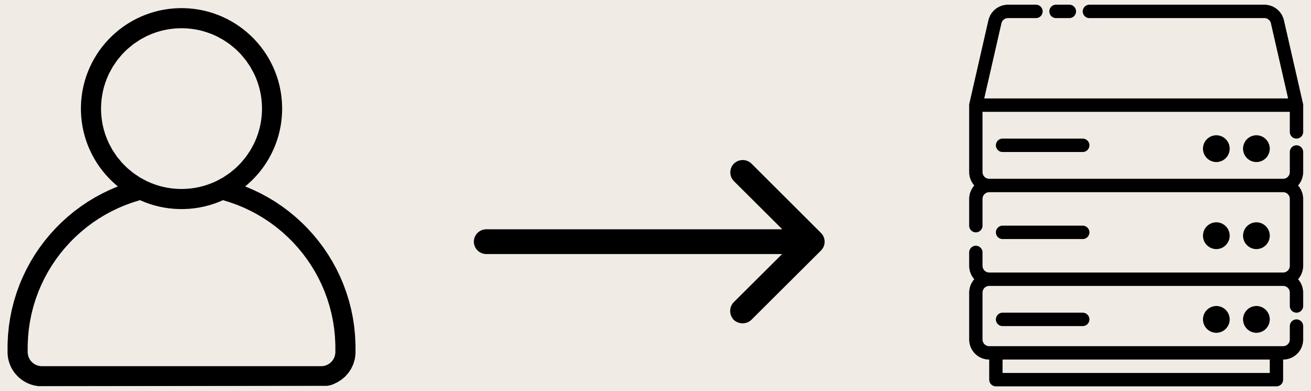
- PKCS #1 v1.5
- RSA Asym key
- SHA- 256 hashing

SECURE DOCUMENT FORMAT

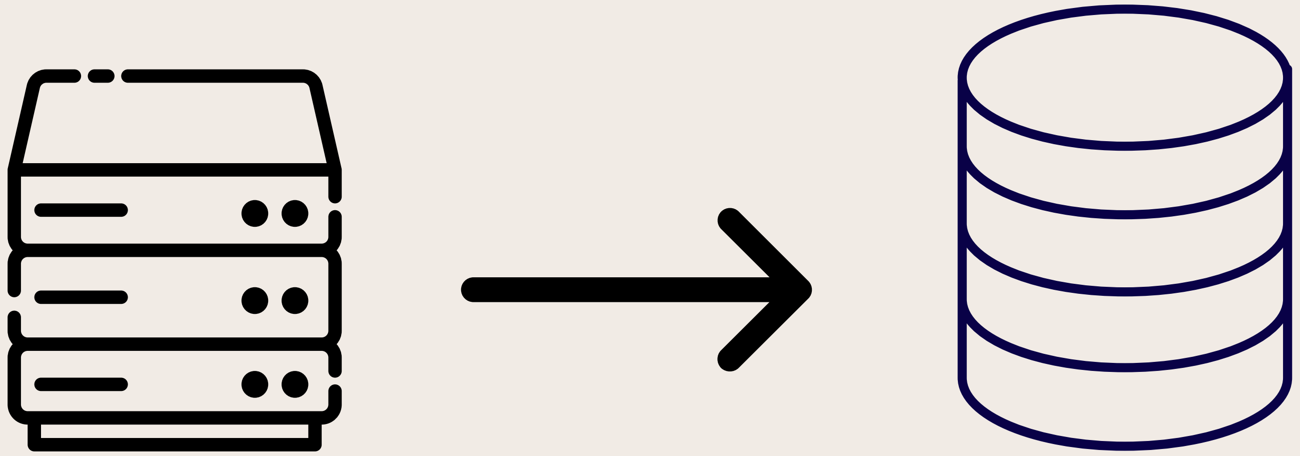


```
encrypted_document = {
    'content':       str({
                         'json':                  json_object or str,
                         'timestamp':             seconds in float with microsecond precision,
                         'nonce':                 str,
                         'encrypted_sections':    list,
                         'fully_encrypted':       bool
                     }),
    'encrypted_key': base64(rsa_encrypt(AES_key + AES_IV)), # optional, [+] concatenates
    'signature':     base64(rsa_sign(sha256(content))),
}
```

SECURITY OF CONNECTIONS & KEY DESTRIBUTION

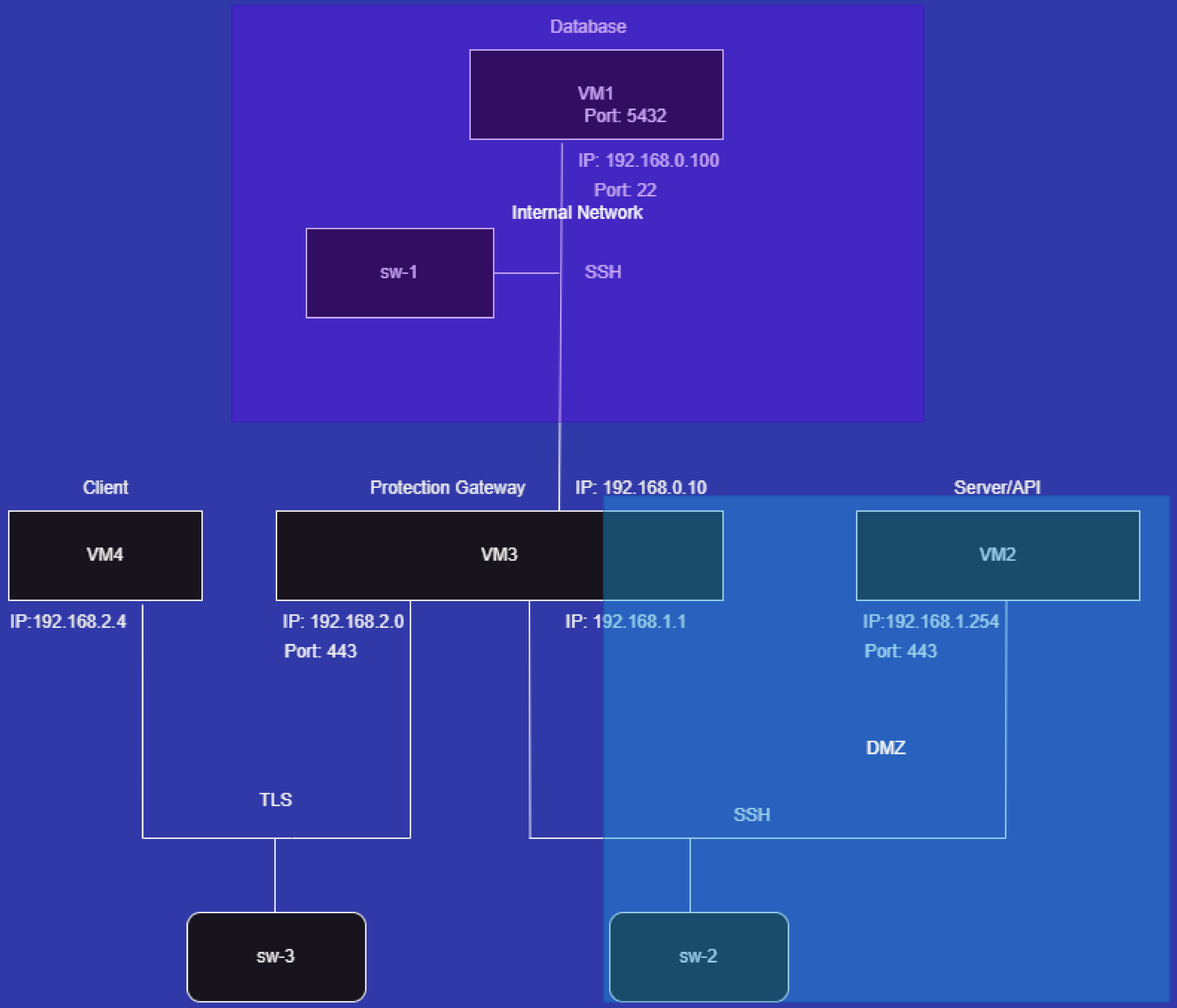


- TLS
- Same client key and certificate for connections
- Shared Server certificate and server public key
- Client share public key to server



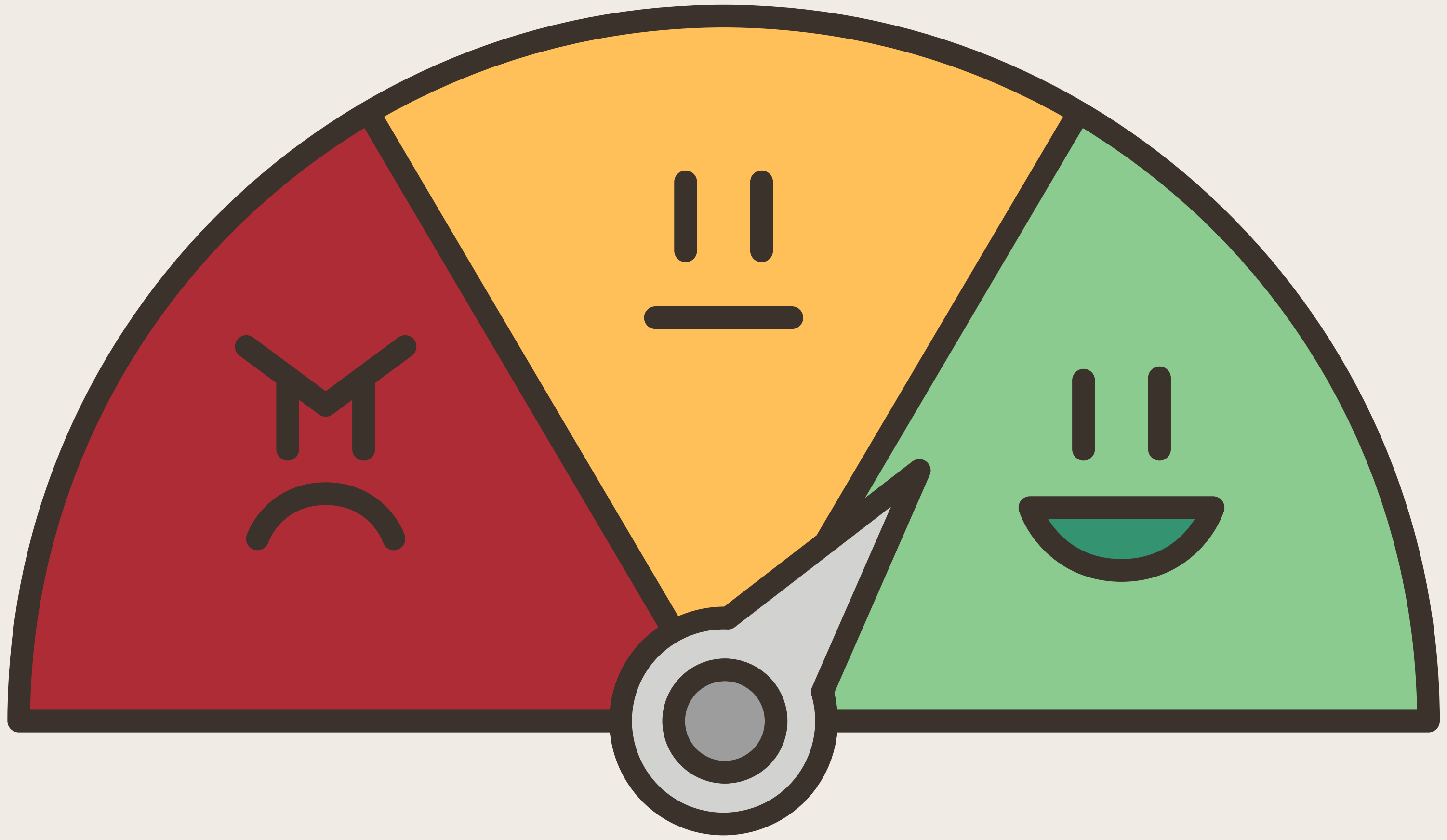
- SSH
- Server knows credentials of database
- Database stores clients public keys
- Database configured with a selected ip

INFRASTRUCTURE & FIREWALLS



SECURITY CHALLENGE

- Review is his own json
- Executing signing just for the review and store the json secure document on the database as the review
- Encrypt function handles all matters of security of the document
- Review added to the json file of the restaurant when read request is made
- Verify review by getting all public keys stores in the database
- Freshness is not verified for the review



MAIN RESULT & CONCLUSIONS

- Constructed the entire infrastructure for a simple service
- Maintained and fortified the machines
- Designed and connected the network layout
- Secured connections with off-the-shelf solutions
- Created customized protocols for added restrictions and assurance
- Validated each message for security
- While meeting most requirements, some aspects fell short, notably the user experience's simplicity and the server's minimal error handling for multi-user interactions

PRESENTED BY LUÍS MARQUES Nº 99265 & RAMIRO MOLDES Nº 99313

THANK YOU VERY MUCH!