Restaurants & Tourism: BombAppetit

a SIRS presentation

Group 57
Duarte Jeremias, 96857
Francisco António, 105741
Rodrigo Liu, 96909

Introduction

- BombAppetit is a web application tailored to enhance the dining experience
- Users can look up any restaurant information
- Restaurants may create discount vouchers to reward loyal customers
- Customers may leave a review

Initial Core Data

Restaurant Info

- Owner (string);
- Name (string);
- Address (string);
- Genre (string);
- Menu (list<item>);

Vouchers

- Code (string);
- Description (string);

Final Core Data (after Security Challenge considerations)

Restaurant Info

- Owner (string);
- Name (string);
- Address (string);
- Genre (string);
- Menu (list<item>);
- *Username (string);*

Vouchers

- Code (string);
- Description (string);
- *Owner (string);*
- *previousOwner (string);*
- *Restaurant (string);*

Reviews

- *Username (string);*
- *Restaurant (string);*
- *Rating (int);*
- *Comment (string);*

Secure Document Algorithms

Confidentiality

- RSA/ECB/PKCS1Padding;
- Only used for the vouchers;
- Uses the owner of the voucher’s public key;

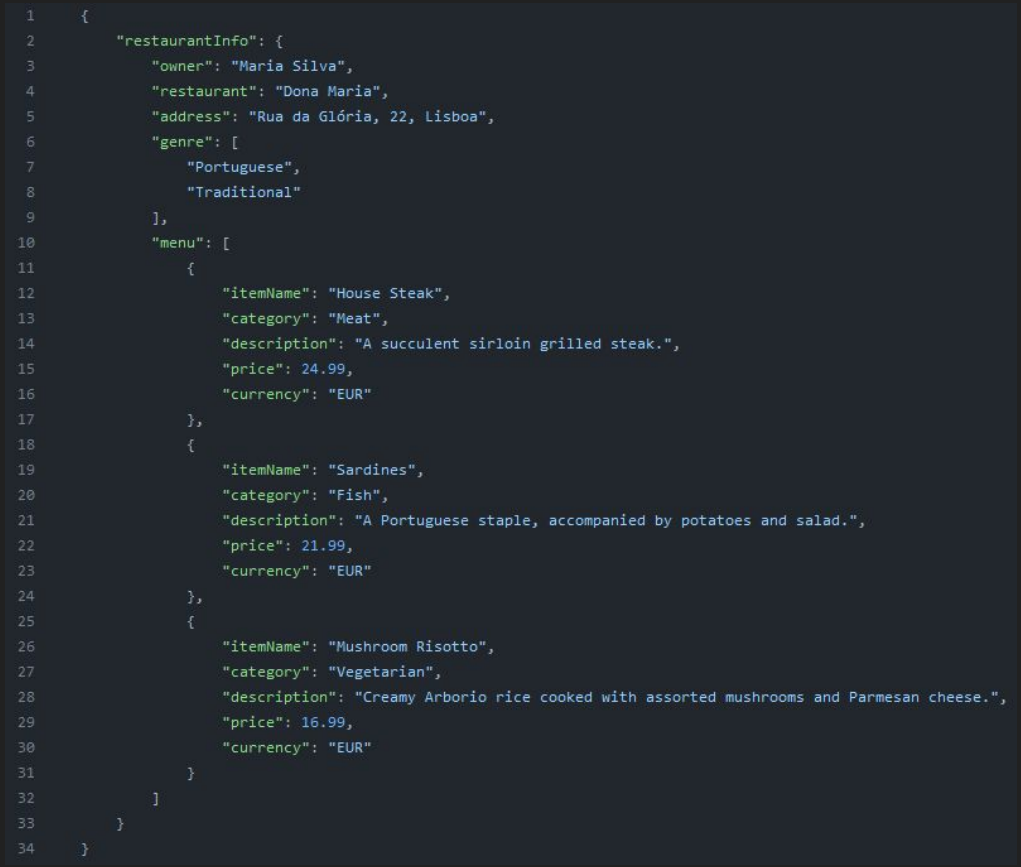
Freshness

- timestamp;
- nonce;

Integrity & Authenticity

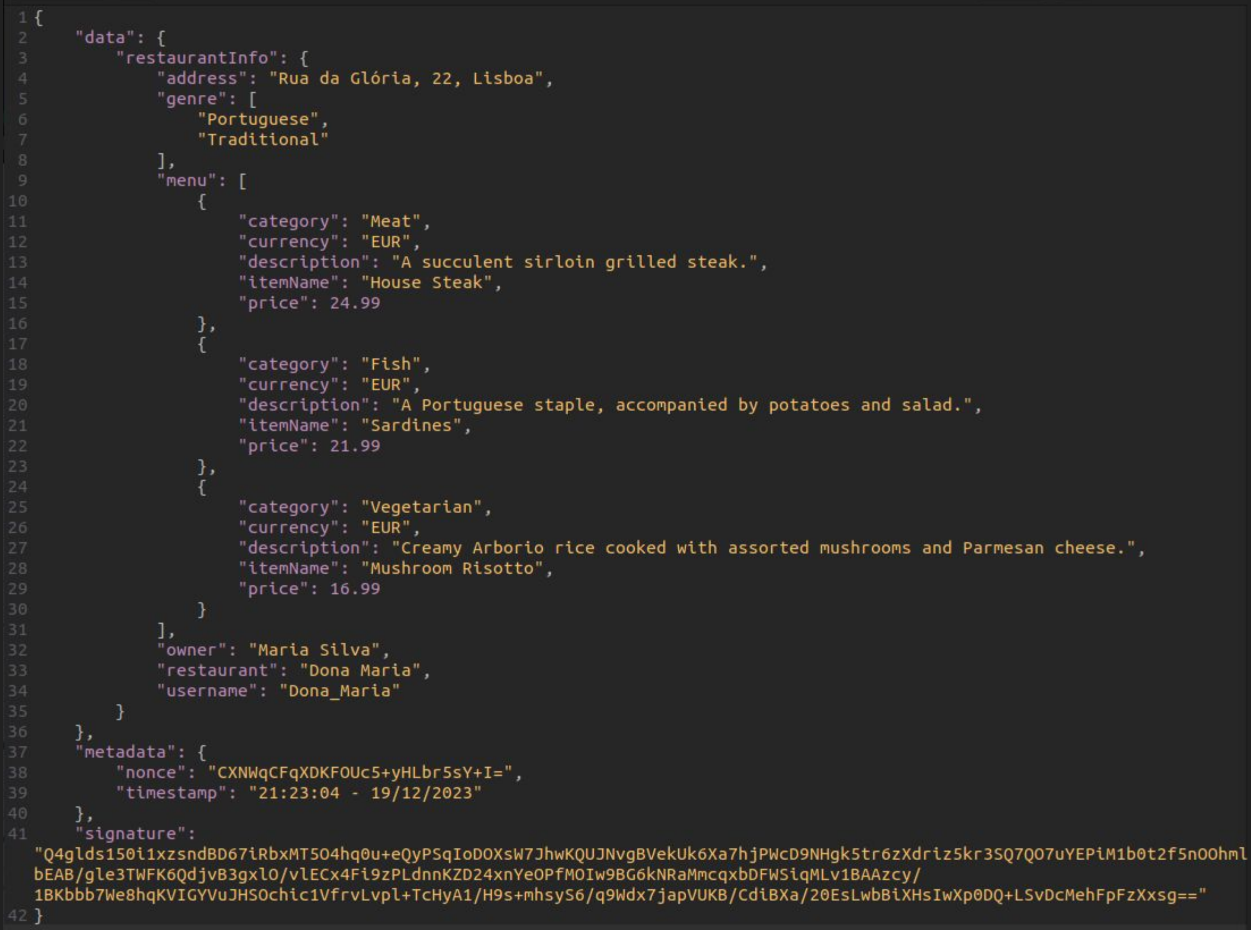
- SHA256withRSA digital signature;
- Uses a SHA256 digest of the whole file;

Secure Document Format - Unprotected Restaurant Info



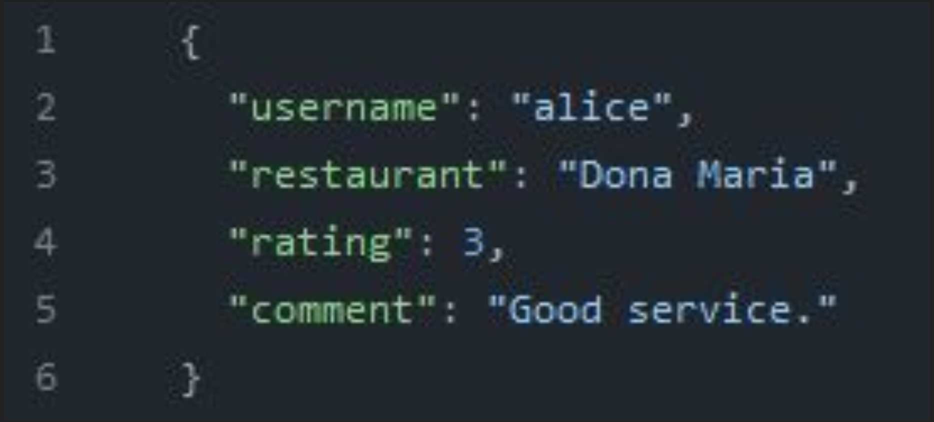
```json
{
    "restaurantInfo": {
        "owner": "Maria Silva",
        "restaurant": "Dona Maria",
        "address": "Rua da Glória, 22, Lisboa",
        "genre": [
            "Portuguese",
            "Traditional"
        ],
        "menu": [
            {
                "itemName": "House Steak",
                "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"
            }
        ]
    }
}
```

Secure Document Format - Protected Restaurant Info



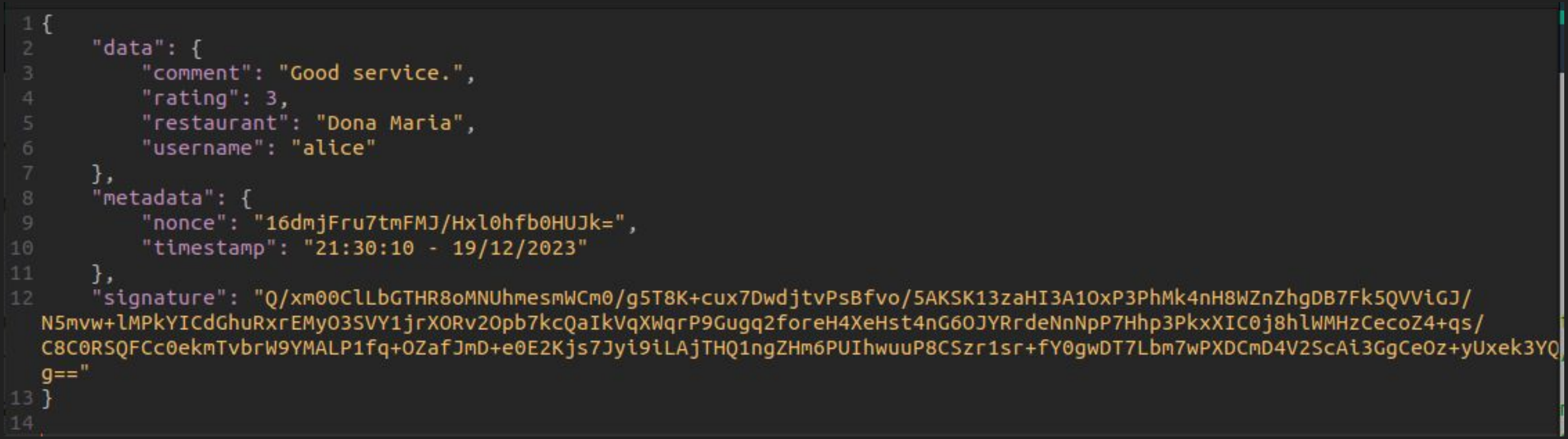
```
{
    "data": {
        "restaurantInfo": {
            "address": "Rua da Glória, 22, Lisboa",
            "genre": [
                "Portuguese",
                "Traditional"
            ],
            "menu": [
                {
                    "category": "Meat",
                    "currency": "EUR",
                    "description": "A succulent sirloin grilled steak.",
                    "itemName": "House Steak",
                    "price": 24.99
                },
                {
                    "category": "Fish",
                    "currency": "EUR",
                    "description": "A Portuguese staple, accompanied by potatoes and salad.",
                    "itemName": "Sardines",
                    "price": 21.99
                },
                {
                    "category": "Vegetarian",
                    "currency": "EUR",
                    "description": "Creamy Arborio rice cooked with assorted mushrooms and Parmesan cheese.",
                    "itemName": "Mushroom Risotto",
                    "price": 16.99
                }
            ],
            "owner": "Maria Silva",
            "restaurant": "Dona Maria",
            "username": "Dona_Maria"
        }
    },
    "metadata": {
        "nonce": "CXNWqCFqXDKFOUc5+yHLbr5sY+I=",
        "timestamp": "21:23:04 - 19/12/2023"
    },
    "signature":
"Q4glds150i1xzsndBD67iRbxMT5O4hq0u+eQyPSqIoDOXsW7JhwKQUJNvgBVekUk6Xa7hjPWcD9NHgk5tr6zXdriz5kr3SQ7QO7uYEPiM1b0t2f5nOOhml
bEAB/gle3TWFK6QdjvB3gxlO/vlECx4Fi9zPLdnnKZD24xnYeOPfMOIw9BG6kNRaMmcqxbDFWSiqMLv1BAAzcy/
1BKbbb7We8hqKVIGYVuJHSOchic1VfrvLvpl+TcHyA1/H9s+mhsyS6/q9Wdx7japVUKB/CdiBXa/20EsLwbBiXHsIwXp0DQ+LSvDcMehFpFzXxsg=="
}
```

Secure Document Format - Unprotected Review



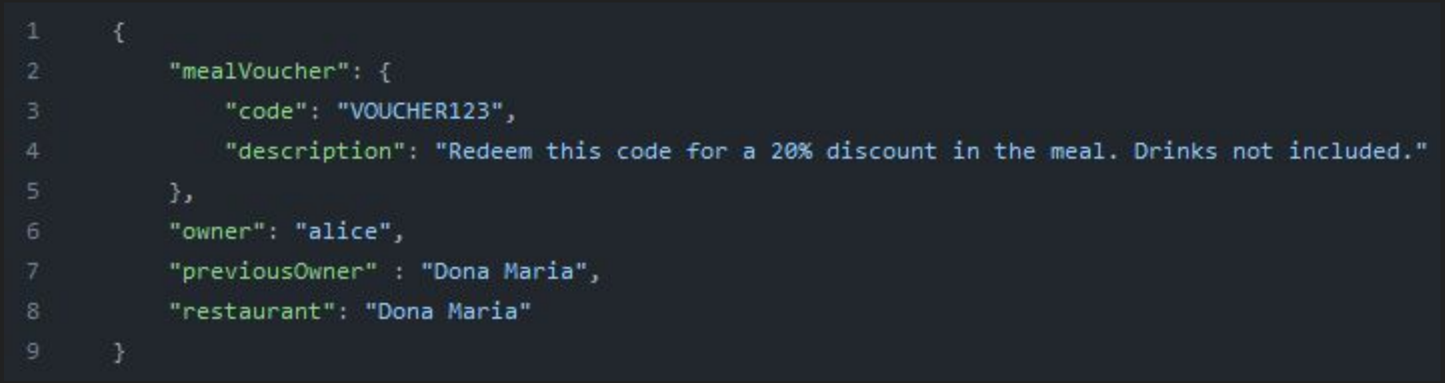
```
{
  "username": "alice",
  "restaurant": "Dona Maria",
  "rating": 3,
  "comment": "Good service."
}
```

Secure Document Format - Protected Review



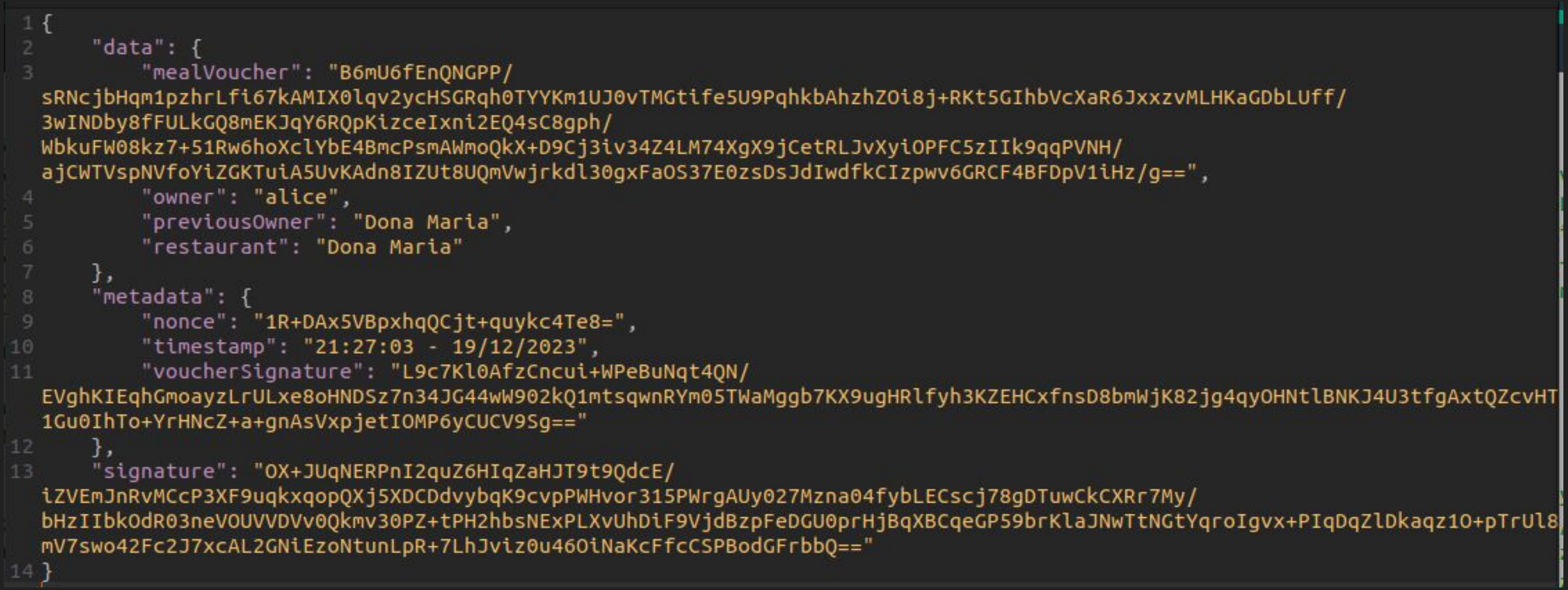
```
{
    "data": {
        "comment": "Good service.",
        "rating": 3,
        "restaurant": "Dona Maria",
        "username": "alice"
    },
    "metadata": {
        "nonce": "16dmjFru7tmFMJ/Hxl0hfb0HUJk=",
        "timestamp": "21:30:10 - 19/12/2023"
    },
    "signature": "Q/xm00ClLbGTHR8oMNUhmesmWCm0/g5T8K+cux7DwdjtvPsBfvo/5AKSK13zaHI3A1OxP3PhMk4nH8WZnZhgDB7Fk5QVViGJ/
N5mvw+lMPkYICdGhuRxrEMyO3SVY1jrXORv2Opb7kcQaIkVqXWqrP9Gugq2foreH4XeHst4nG6OJYRrdeNnNpP7Hhp3PkxXIC0j8hlWMHzCecoZ4+qs/
C8C0RSQFCc0ekmTvbrW9YMALP1fq+OZafJmD+e0E2Kjs7Jyi9iLAjTHQ1ngZHm6PUIhwuuP8CSzr1sr+fY0gwDT7Lbm7wPXDCmD4V2ScAi3GgCeOz+yUxek3YQ
g=="
}
```

Secure Document Format - Unprotected Voucher



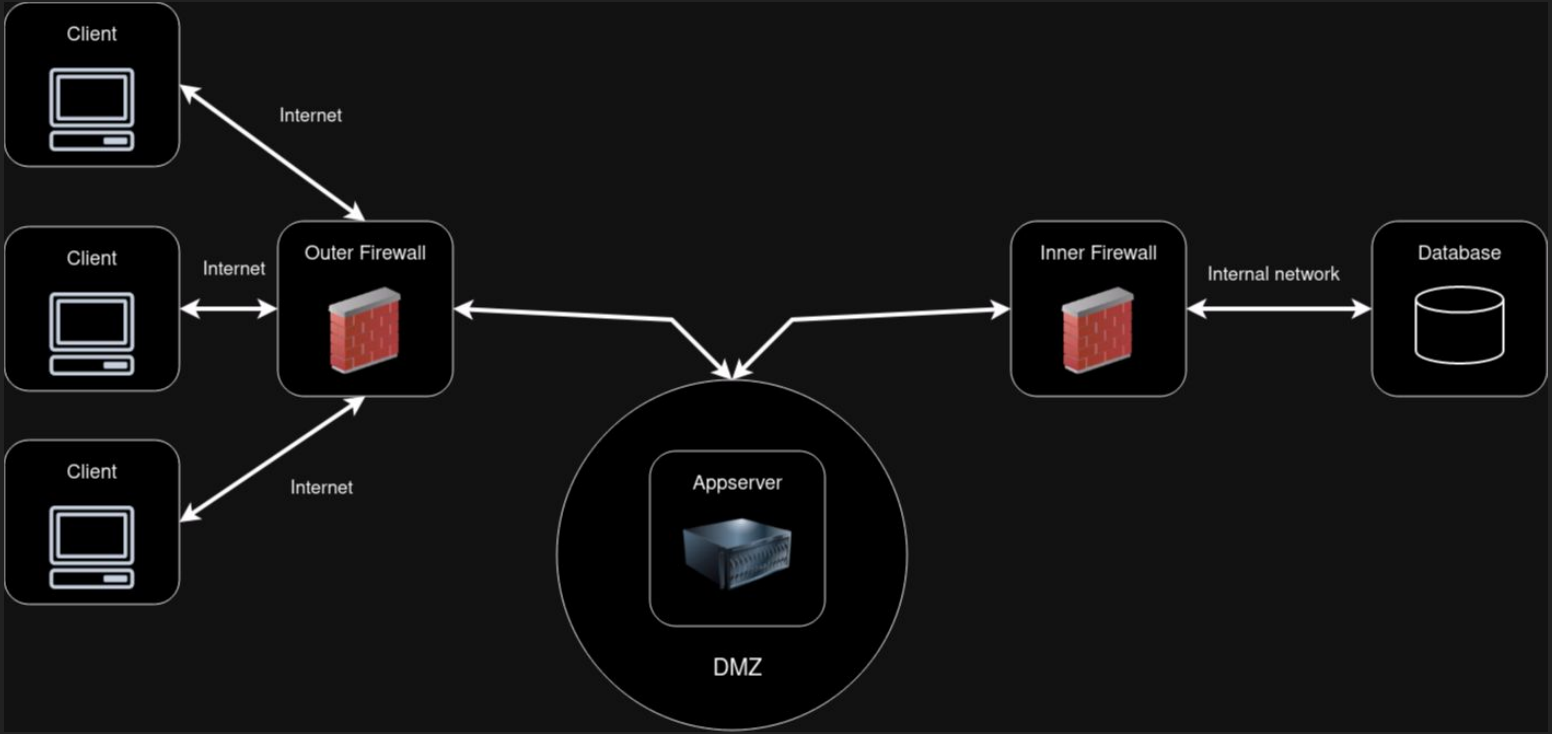
```
{
    "mealVoucher": {
        "code": "VOUCHER123",
        "description": "Redeem this code for a 20% discount in the meal. Drinks not included."
    },
    "owner": "alice",
    "previousOwner" : "Dona Maria",
    "restaurant": "Dona Maria"
}
```

Secure Document Format - Protected Voucher

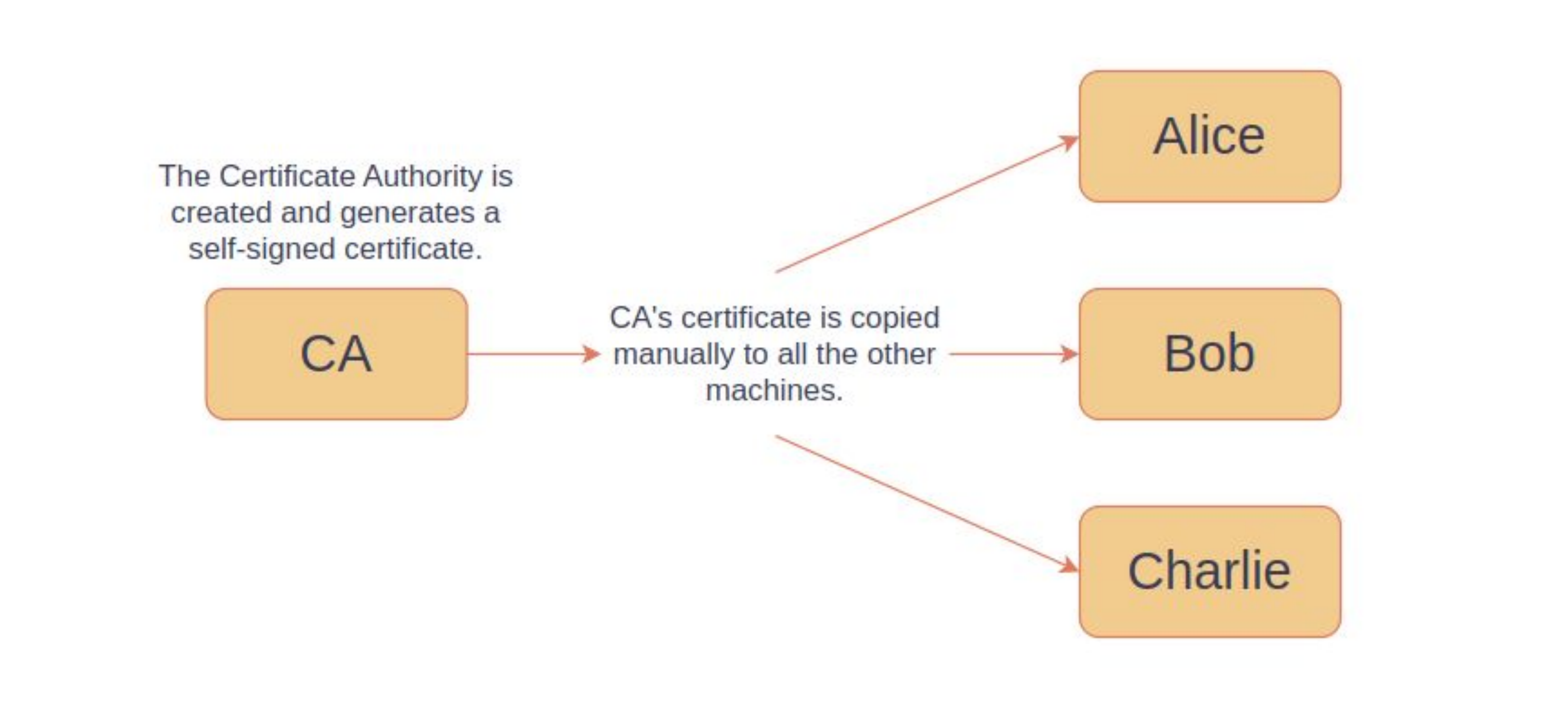


```
{
    "data": {
        "mealVoucher": "B6mU6fEnQNGPP/
sRNcjbHqm1pzhrLfi67kAMIX0lqv2ycHSGRqh0TYYKm1UJ0vTMGtife5U9PqhkbAhzhZOi8j+RKt5GIhbVcXaR6JxxzvMLHKaGDbLUff/
3wINDby8fFULkGQ8mEKJqY6RQpKizceIxni2EQ4sC8gph/
WbkuFW08kz7+51Rw6hoXclYbE4BmcPsmAWmoQkX+D9Cj3iv34Z4LM74XgX9jCetRLJvXyiOPFC5zIIk9qqPVNH/
ajCWTVspNVfoYiZGKTuiA5UvKAdn8IZUt8UQmVwjrkdl30gxFaOS37E0zsDsJdIwdfkCIzpwv6GRCF4BFDpV1iHz/g==",
        "owner": "alice",
        "previousOwner": "Dona Maria",
        "restaurant": "Dona Maria"
    },
    "metadata": {
        "nonce": "1R+DAx5VBpxhqQCjt+quykc4Te8=",
        "timestamp": "21:27:03 - 19/12/2023",
        "voucherSignature": "L9c7Kl0AfzCncui+WPeBuNqt4QN/
EVghKIEqhGmoayzLrULxe8oHNDSz7n34JG44wW902kQ1mtsqwnRYm05TWaMggb7KX9ugHRlfyh3KZEHCxfnsD8bmWjK82jg4qyOHNtlBNKJ4U3tfgAxtQZcvHT
1Gu0IhTo+YrHNcZ+a+gnAsVxpjetIOMP6yCUCV9Sg=="
    },
    "signature": "OX+JUqNERPnI2quZ6HIqZaHJT9t9QdcE/
iZVEmJnRvMCcP3XF9uqkxqopQXj5XDCDdvybqK9cvpPWHvor315PWrgAUy027Mzna04fybLECscj78gDTuwCkCXRr7My/
bHzIIbkOdR03neVOUVVDVv0Qkmv30PZ+tPH2hbsNExPLXvUhDiF9VjdBzpFeDGU0prHjBqXBCqeGP59brKlaJNwTtNGtYqroIgvx+PIqDqZlDkaqz1O+pTrUl8
mV7swo42Fc2J7xcAL2GNiEzoNtunLpR+7LhJviz0u46OiNaKcFfcCSPBodGFrbbQ=="
}
```

Infrastructure



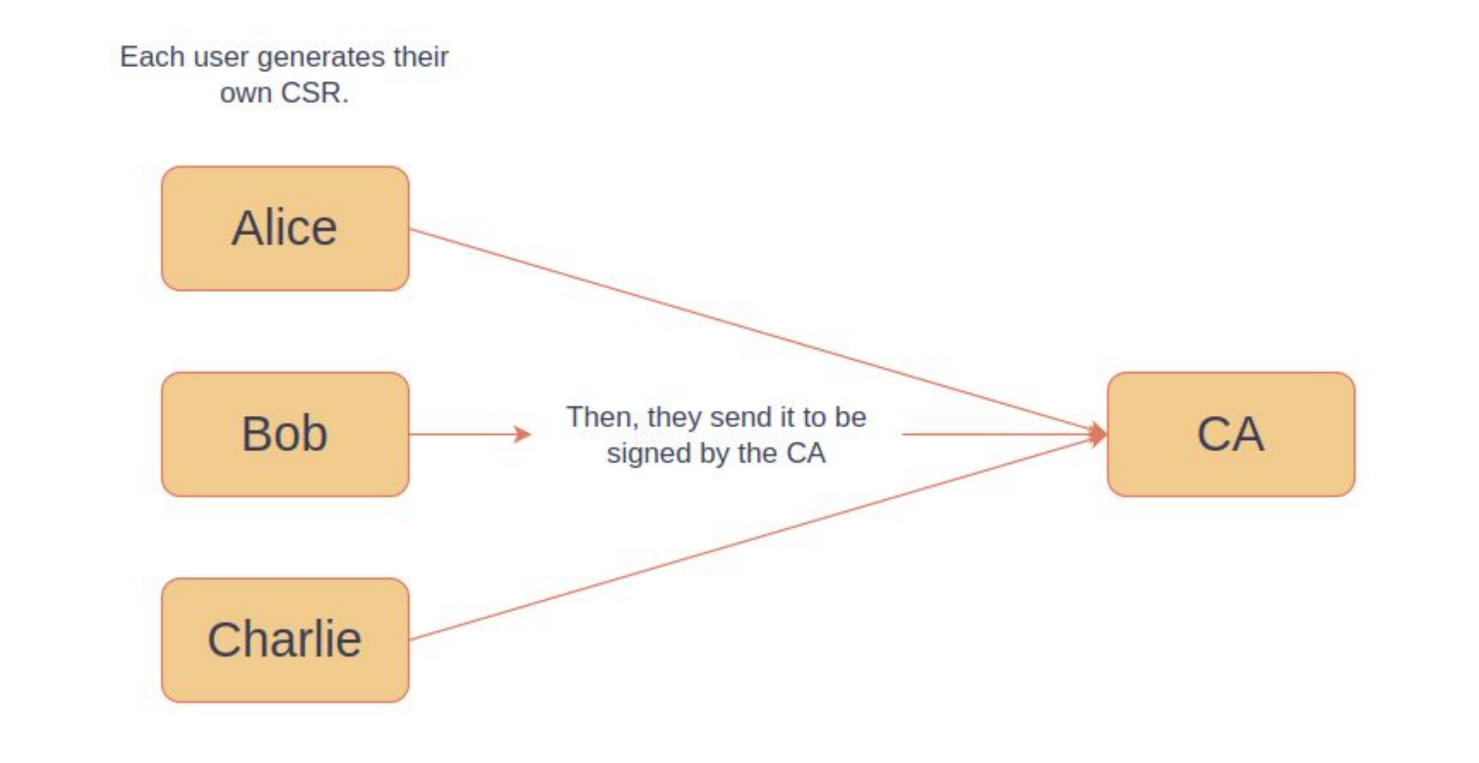
Key Distribution - 1



The Certificate Authority is created and generates a self-signed certificate.

CA's certificate is copied manually to all the other machines.

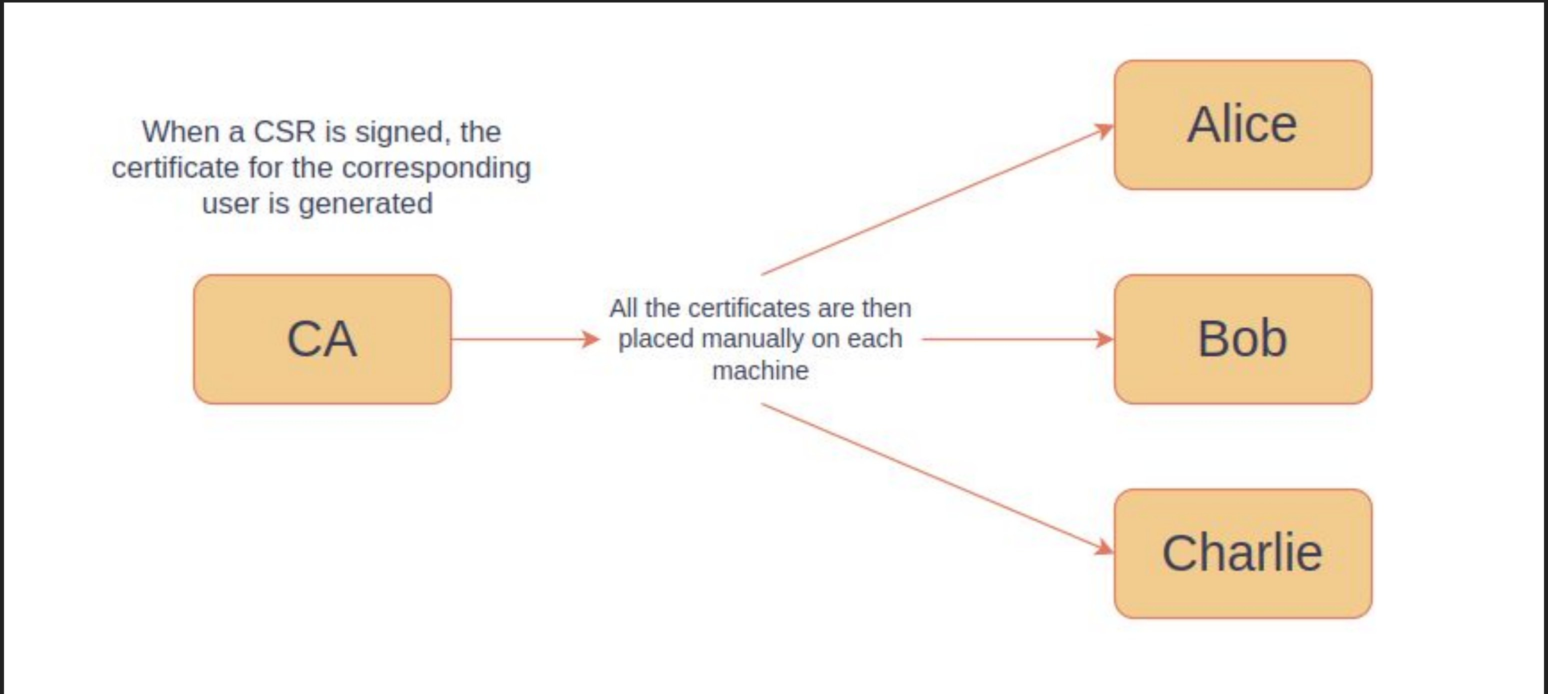
Key Distribution - 2



Each user generates their own CSR.

Then, they send it to be signed by the CA

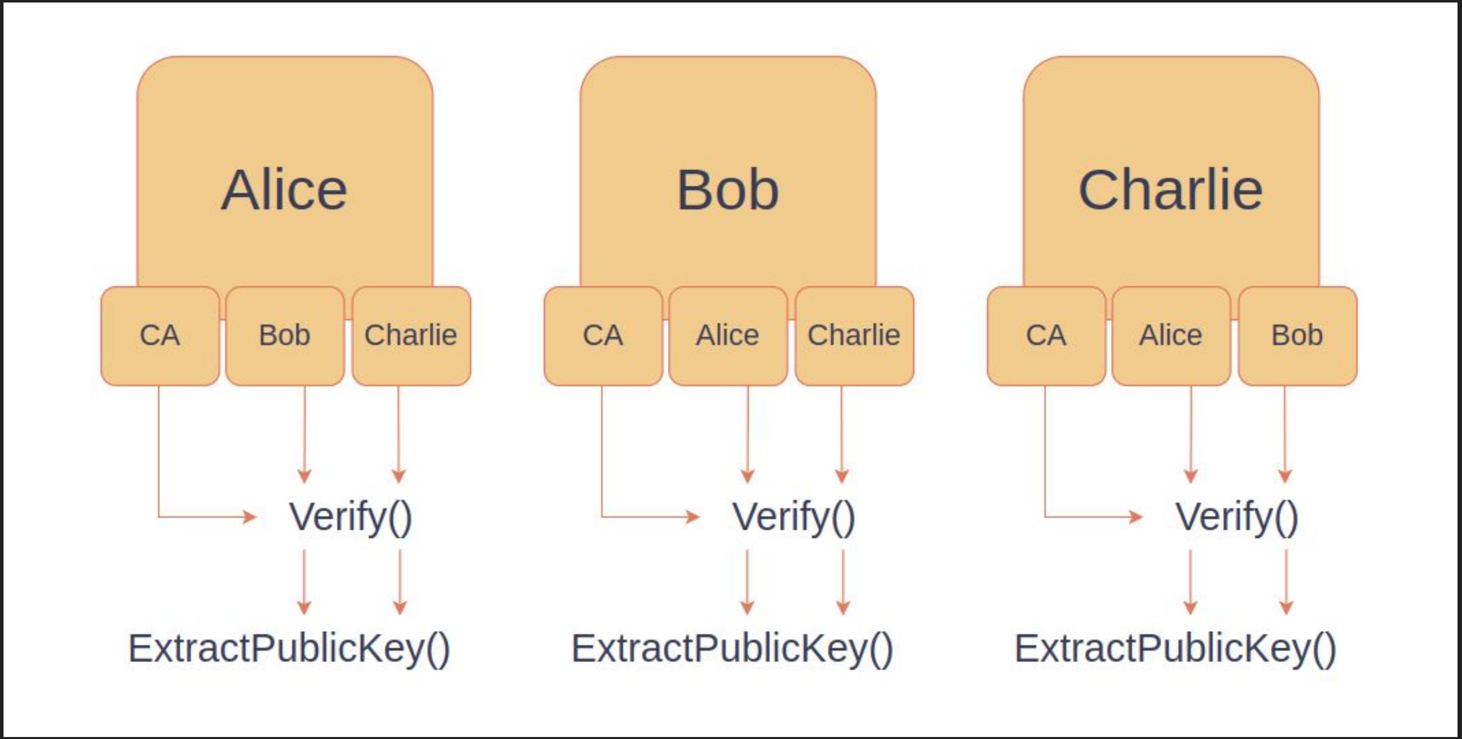
Key Distribution - 3



When a CSR is signed, the certificate for the corresponding user is generated

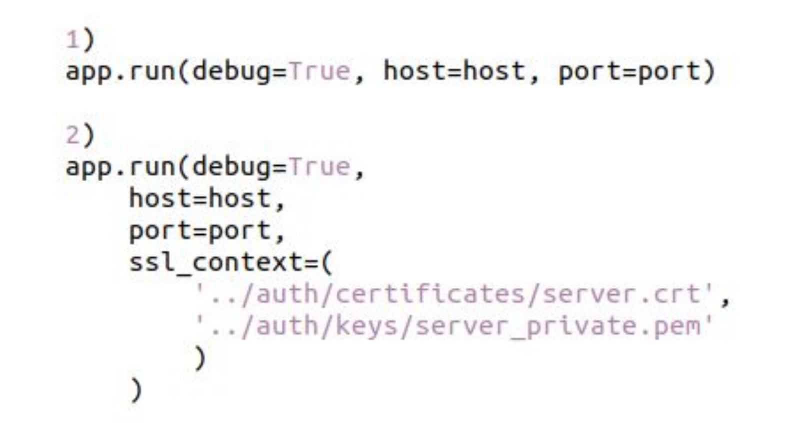
All the certificates are then placed manually on each machine

Key Distribution - 4



Secure Communication

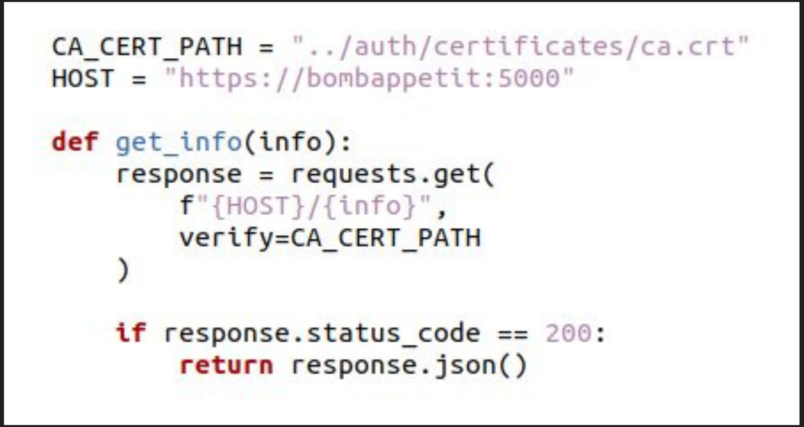
Flask Server



```
1)
app.run(debug=True, host=host, port=port)

2)
app.run(debug=True,
    host=host,
    port=port,
    ssl_context=(
        '../auth/certificates/server.crt',
        '../auth/keys/server_private.pem'
        )
    )
```

Python Client



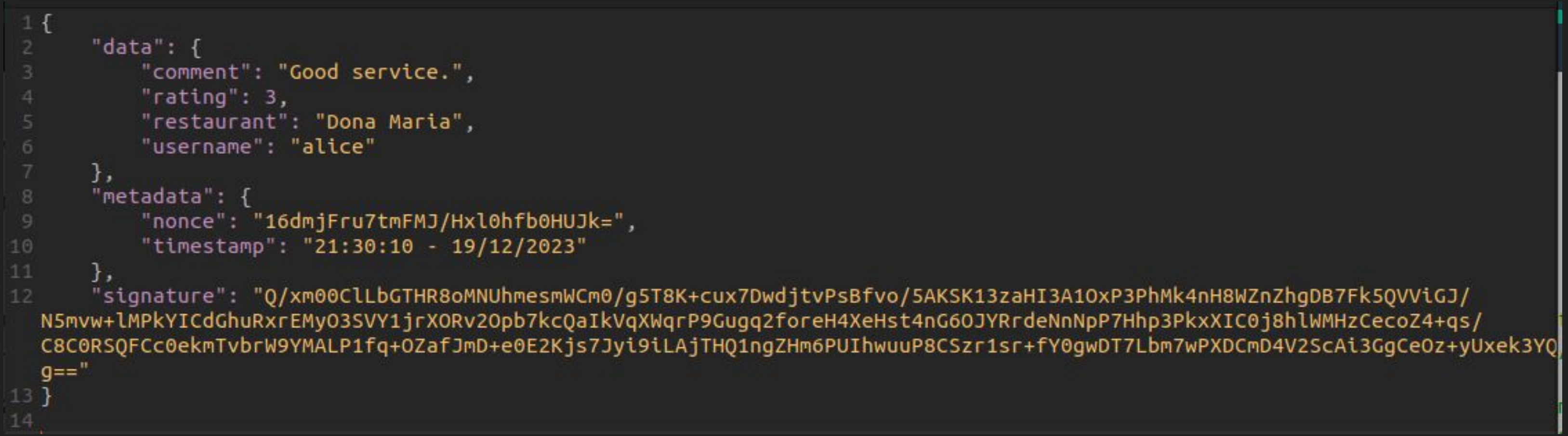
```
CA_CERT_PATH = "../auth/certificates/ca.crt"
HOST = "https://bombappetit:5000"

def get_info(info):
    response = requests.get(
        f"{HOST}/{info}",
        verify=CA_CERT_PATH
    )

    if response.status_code == 200:
        return response.json()
```

Security Challenge - Reviews

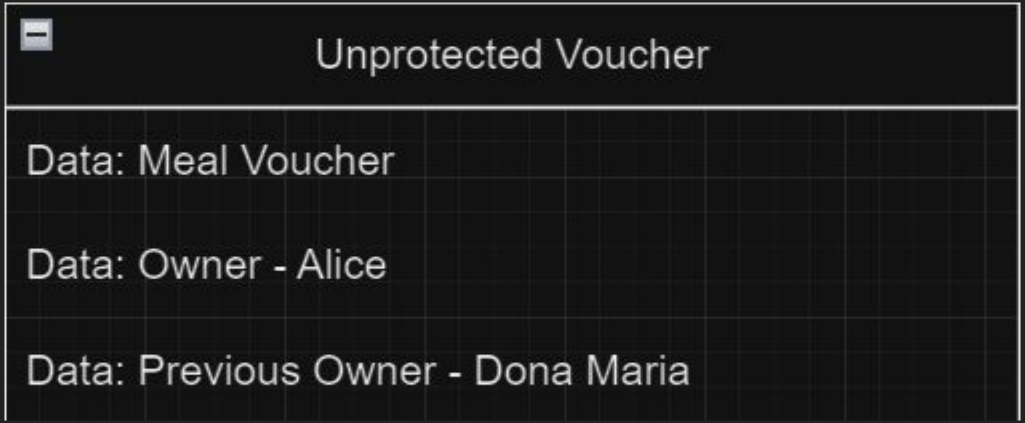
- All reviews are signed by their authors



```
{
    "data": {
        "comment": "Good service.",
        "rating": 3,
        "restaurant": "Dona Maria",
        "username": "alice"
    },
    "metadata": {
        "nonce": "16dmjFru7tmFMJ/Hxl0hfb0HUJk=",
        "timestamp": "21:30:10 - 19/12/2023"
    },
    "signature": "Q/xm00ClLbGTHR8oMNUhmesmWCm0/g5T8K+cux7DwdjtvPsBfvo/5AKSK13zaHI3A1OxP3PhMk4nH8WZnZhgDB7Fk5QVViGJ/
N5mvw+lMPkYICdGhuRxrEMyO3SVY1jrXORv2Opb7kcQaIkVqXWqrP9Gugq2foreH4XeHst4nG6OJYRrdeNnNpP7Hhp3PkxXIC0j8hlWMHzCecoZ4+qs/
C8C0RSQFCc0ekmTvbrW9YMALP1fq+OZafJmD+e0E2Kjs7Jyi9iLAjTHQ1ngZHm6PUIhwuuP8CSzr1sr+fY0gwDT7Lbm7wPXDCmD4V2ScAi3GgCeOz+yUxek3YQ
g=="
}
```

Security Challenge - Vouchers

Unprotected Voucher



Unprotected Voucher

Data: Meal Voucher

Data: Owner - Alice

Data: Previous Owner - Dona Maria

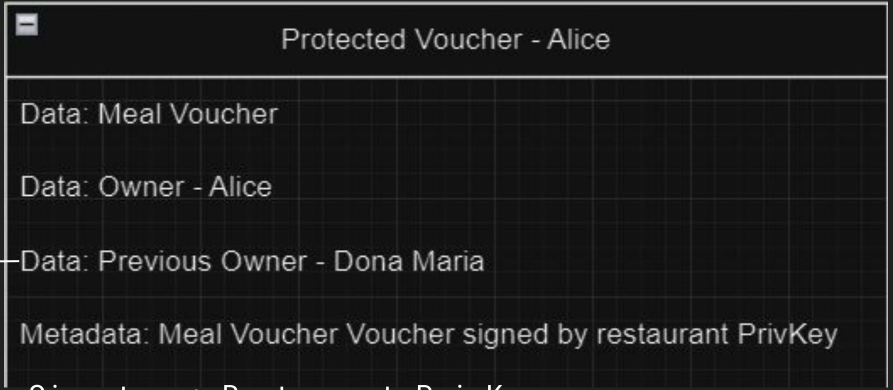
Security Challenge - Vouchers

Voucher Creation

```
secdoc protect --voucher <restaurant's private key> <owner's public key> <unprotected voucher>
```

- Secdoc signs the meal voucher with the restaurant’s private key
- Secdoc encrypts the meal voucher with the owner’s public key
- Secdoc signs the data and metadata together with the restaurant’s private key

Restaurant (Dona Maria) becomes previous owner



Protected Voucher - Alice

Data: Meal Voucher

Data: Owner - Alice

Data: Previous Owner - Dona Maria

Metadata: Meal Voucher Voucher signed by restaurant PrivKey

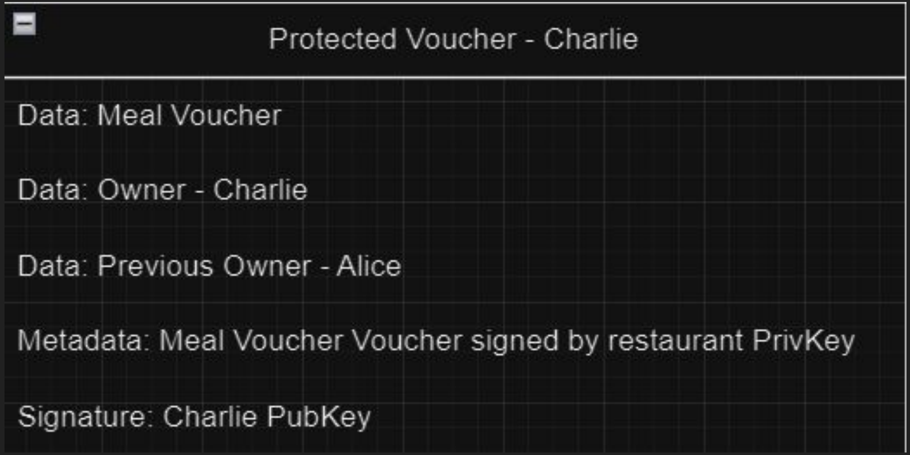
```
Signature: Restaurant PrivKey
```

Security Challenge - Vouchers

Voucher Transfer

- Alice removes encryption
- Changes previous owner to herself
- Changes current owner to Charlie
- Alice encrypts with Charlie's public key
- Alice signs the whole protected voucher including Metadata, with her private key

The contents of the voucher itself remain authentic since its unencrypted signature is the same, and signed by the restaurant, the original issuer of the voucher, so the current owner can always check it for tampering from the other users



Protected Voucher - Charlie

Data: Meal Voucher

Data: Owner - Charlie

Data: Previous Owner - Alice

Metadata: Meal Voucher Voucher signed by restaurant PrivKey

Signature: Charlie PubKey

Demonstration