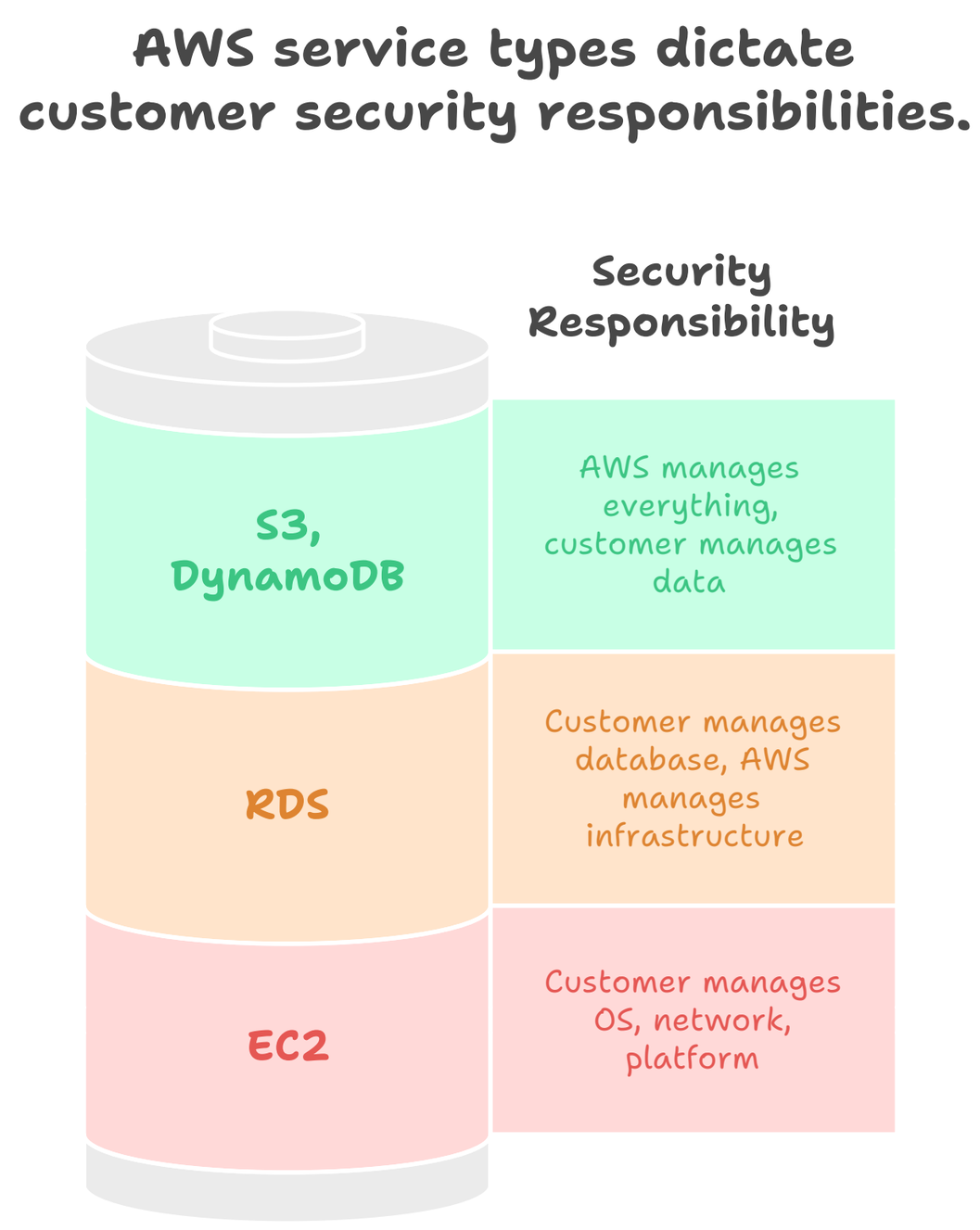
AWS Shared Responsibility Model and S3 Security Summary

Shared Responsibility Model

- Works like different housing options (owning vs. renting vs. hotel)
- Divides security responsibilities between AWS and customers
- Different services have different responsibility splits:
    - EC2 (infrastructure): More customer responsibility
    - Container services (RDS): Shared responsibility
    - Managed services (S3, DynamoDB): AWS handles most responsibilities



S3 Security Layers

1. Network Layer

- S3 is accessible over the internet by default
- Options to block public access
- VPC endpoints provide private connections without using public internet

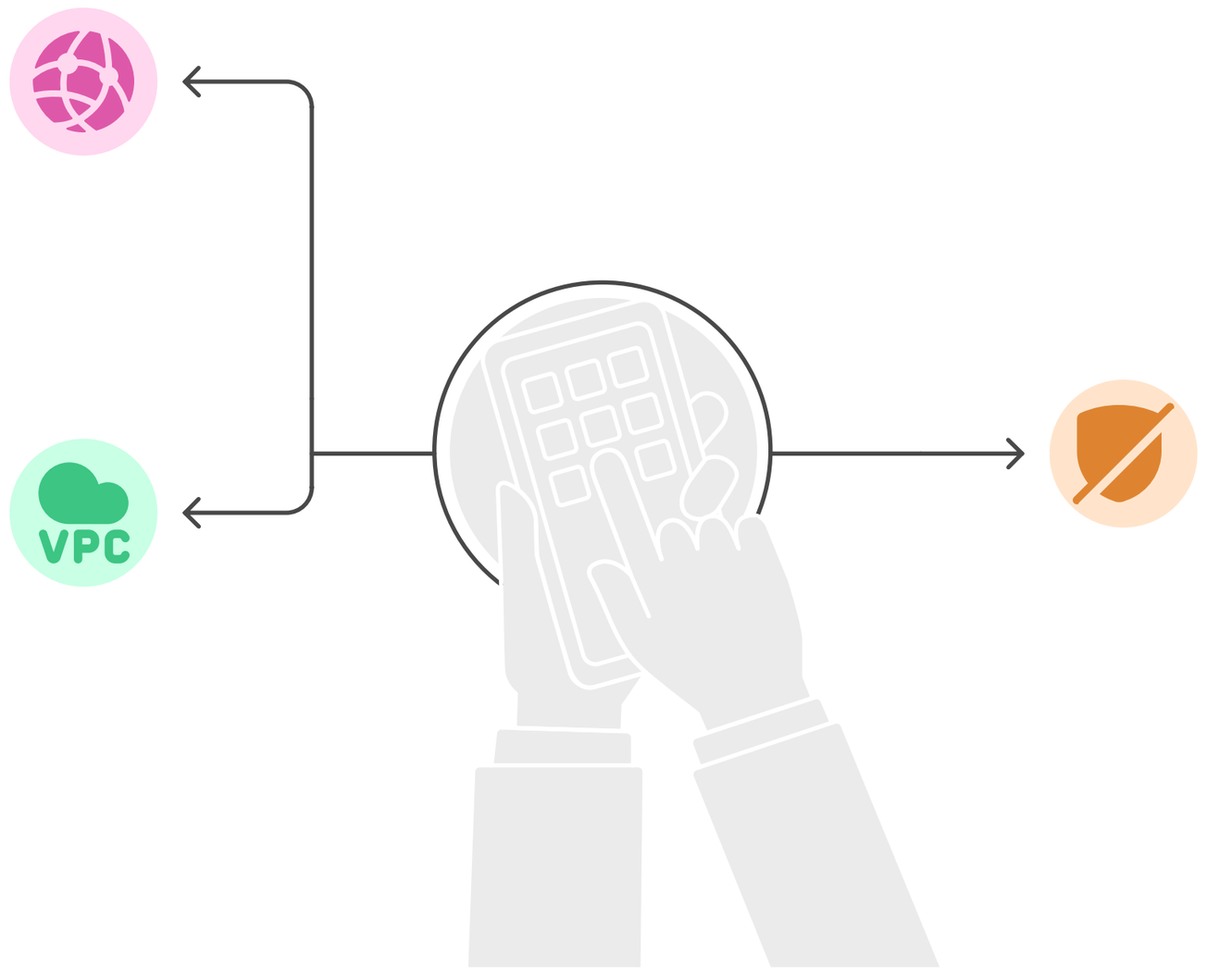
S3 Network Access

Public Accessibility

S3 buckets are, by default, accessible over the internet. This allows for easy access and sharing of data.

VPC Endpoints

VPC endpoints offer private connections to S3. This avoids the public internet, improving security and control over data access.

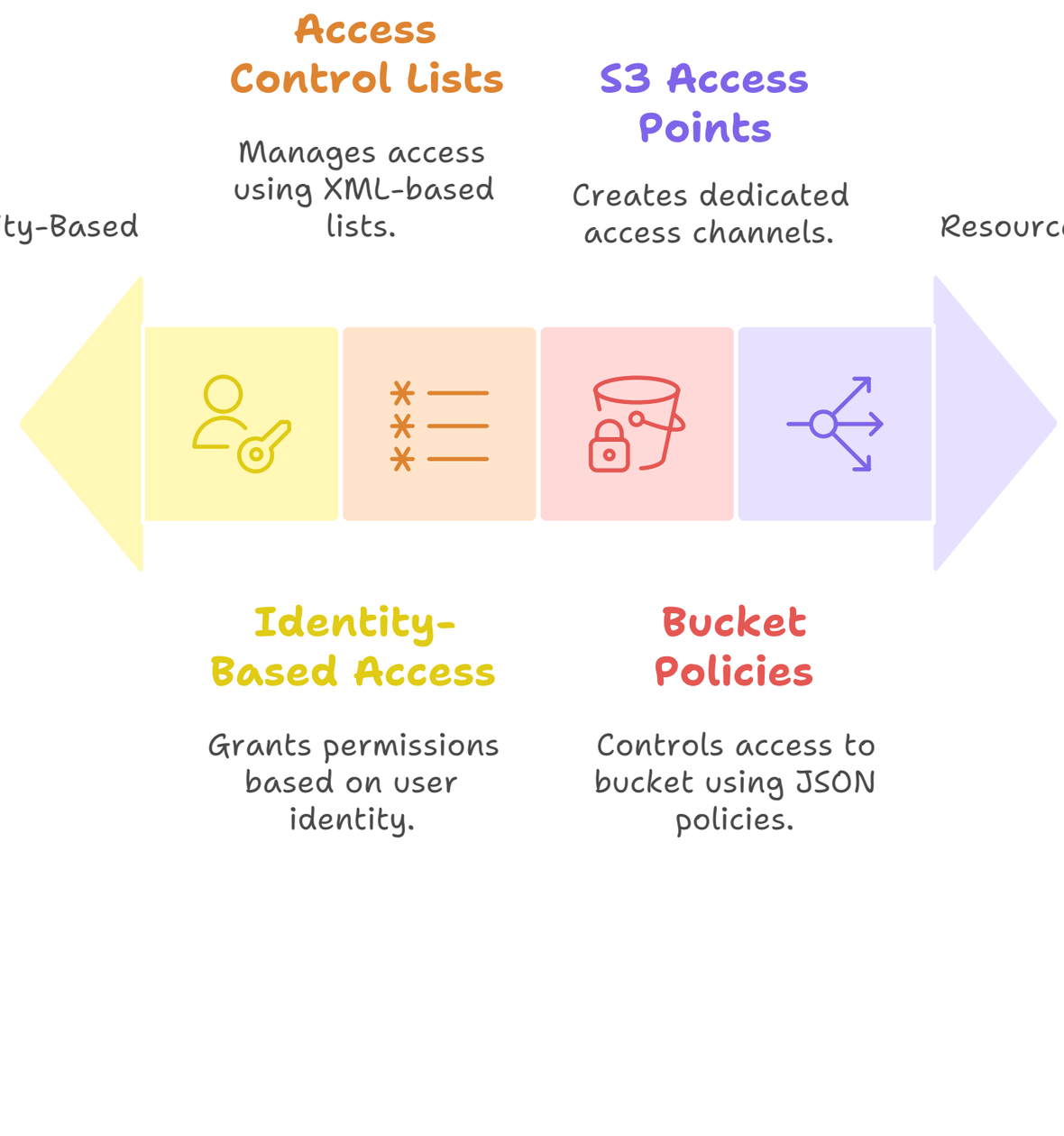


Block Public Access

Options exist to block public access to S3 buckets. This enhances security and restricts access to authorized users only.

2. Access Control Layer

- **Identity-based access**: IAM policies attached to users/groups/roles
- **Resource-based access**:
    - Access Control Lists (ACLs): XML-based, attach to objects/buckets
    - Bucket Policies: JSON-based, control bucket-level access
    - S3 Access Points: Create specific access channels for different applications/teams



3. Encryption Layer

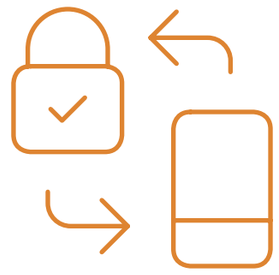
- **Server-side encryption** (at rest):
    - Customer-provided keys: You provide keys, AWS handles encryption
    - AWS Key Management Service options:
        - AWS-owned keys: Free, managed by AWS
        - AWS-managed keys: Stored in your account, managed by AWS, incurs charges
        - Customer-managed keys: You create and manage, highest control, incurs charges
- **In-transit encryption**:
    - Automatic using SSL/TLS protocols
- **Client-side encryption**:
    - Your application encrypts data before sending to AWS
    - AWS never sees unencrypted data

Encryption Types

Server-side Encryption

AWS handles encryption using customer or AWS keys. Different key management options available.

1



In-transit Encryption

Data is automatically encrypted while being transmitted. Uses SSL/TLS protocols for secure communication.

2

Client-side Encryption

Application encrypts data before sending it to AWS. AWS never sees the unencrypted data.

3

IAM responsibilities are shared between customers (configuring permissions) and AWS (securing the IAM service itself).