

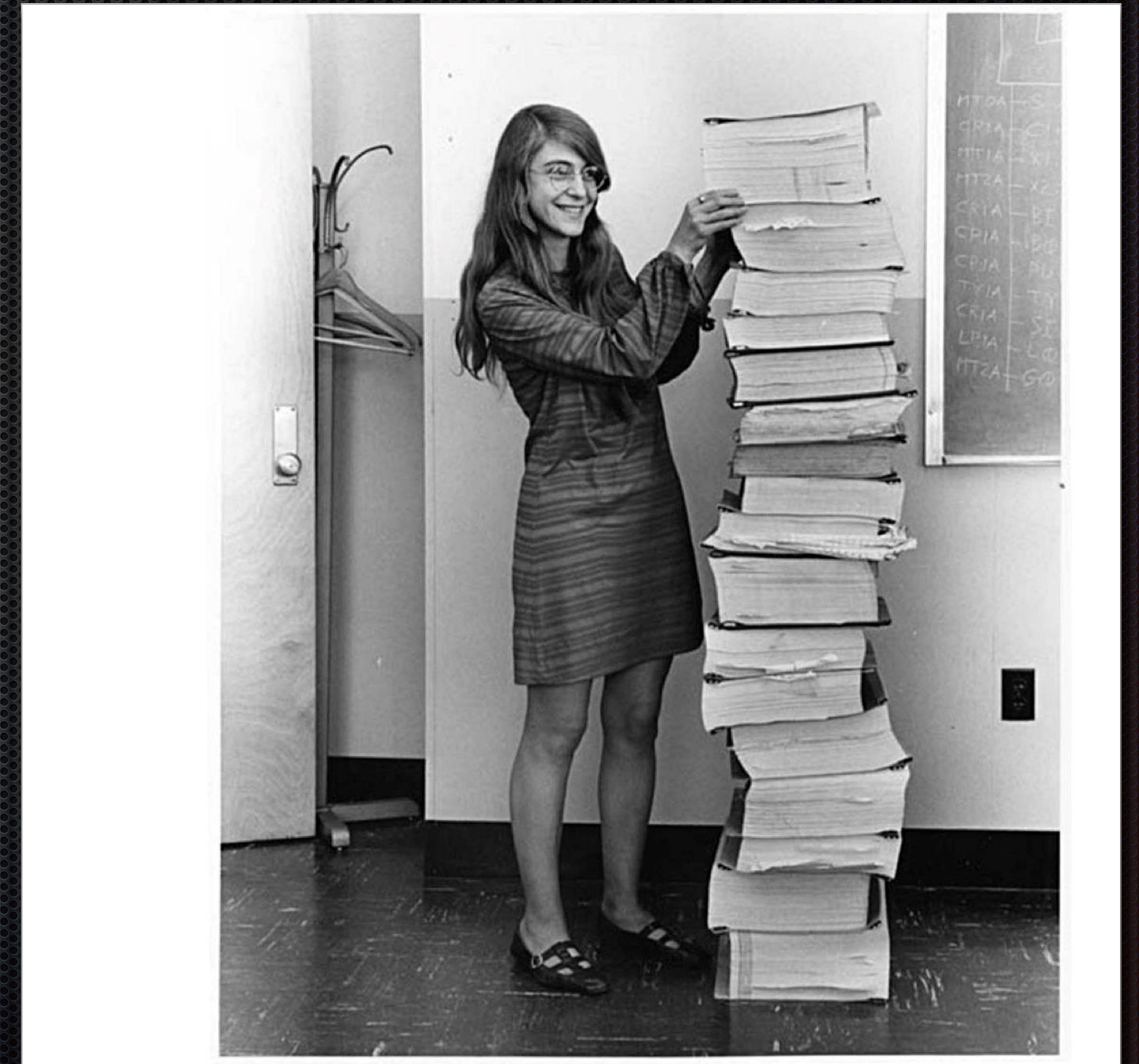
A supercomputer in your hand

David Reidy

Warning: This presentation contains computer code, and Latin

The 1960s

- This is Margaret Hamilton
- That is the code is for the guidance computer that took Apollo 11 to the Moon
- That's a pretty super computer
- It was a mobile computer



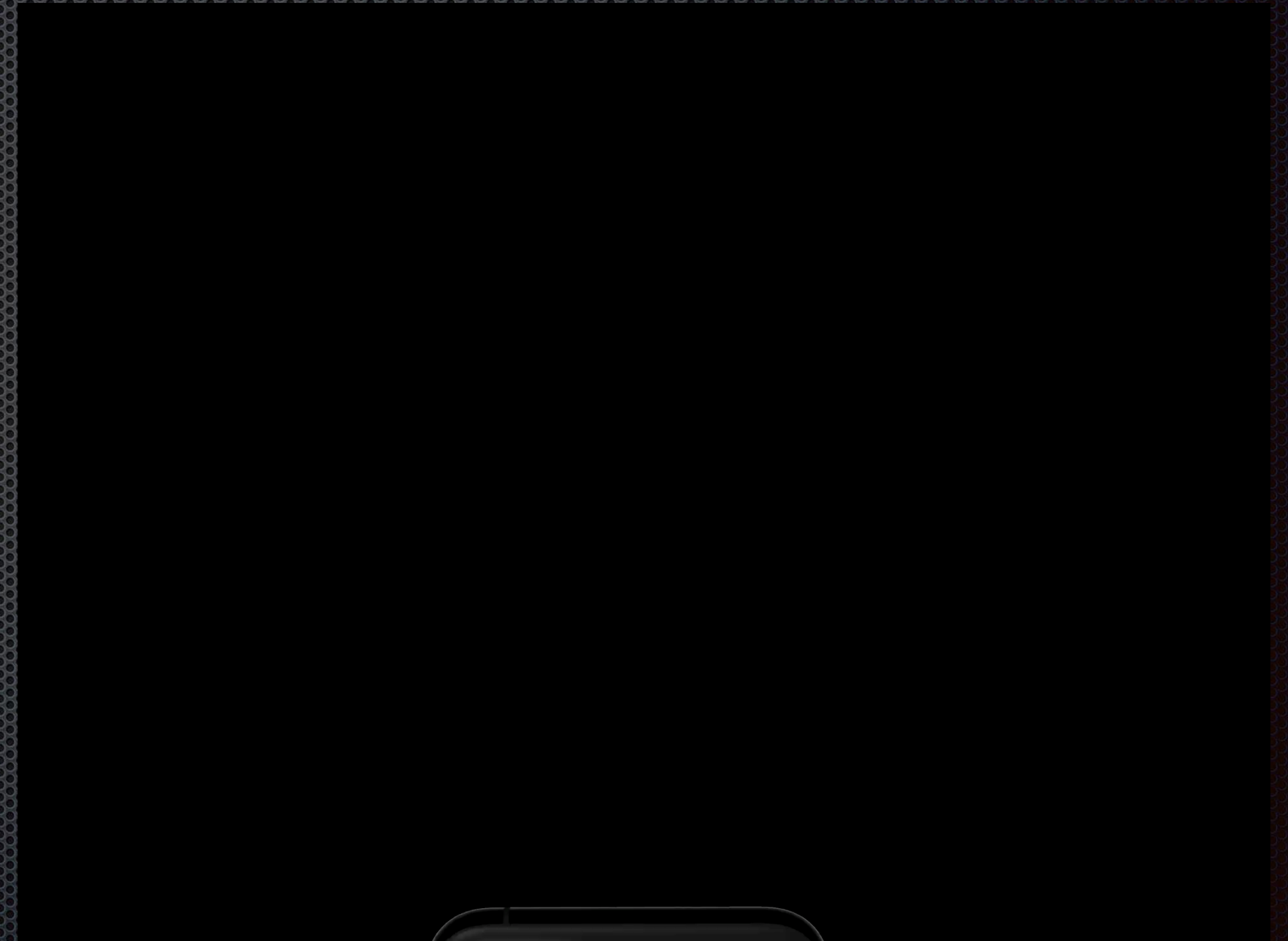
The 1960s

- IBM System/360 - 85
- 4MB RAM
- Around 3 MHz
- 6.5 tonnes (just the processor)
- Computers got their own floors in office buildings



Now

- Almost everyone has something like this
- iPhone 11 Pro Max
 - 4 GB RAM
 - 2 x 2.65 GHz + 4 x 1.8 GHz
 - 226 g
 - Fits in a pocket



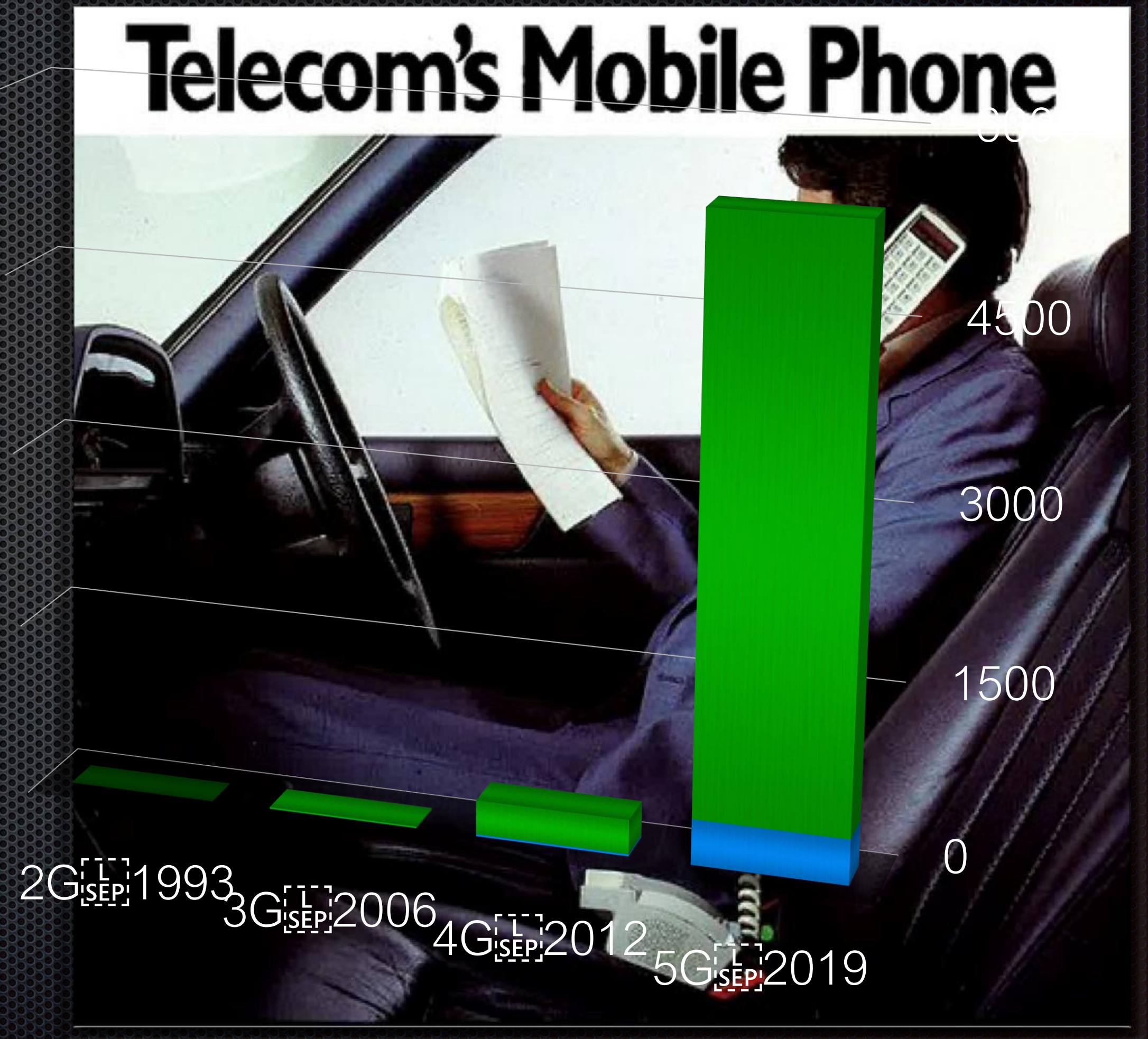
Now

- Not everyone has one of these
- Data centre in North Virginia
- Computers now get their own buildings

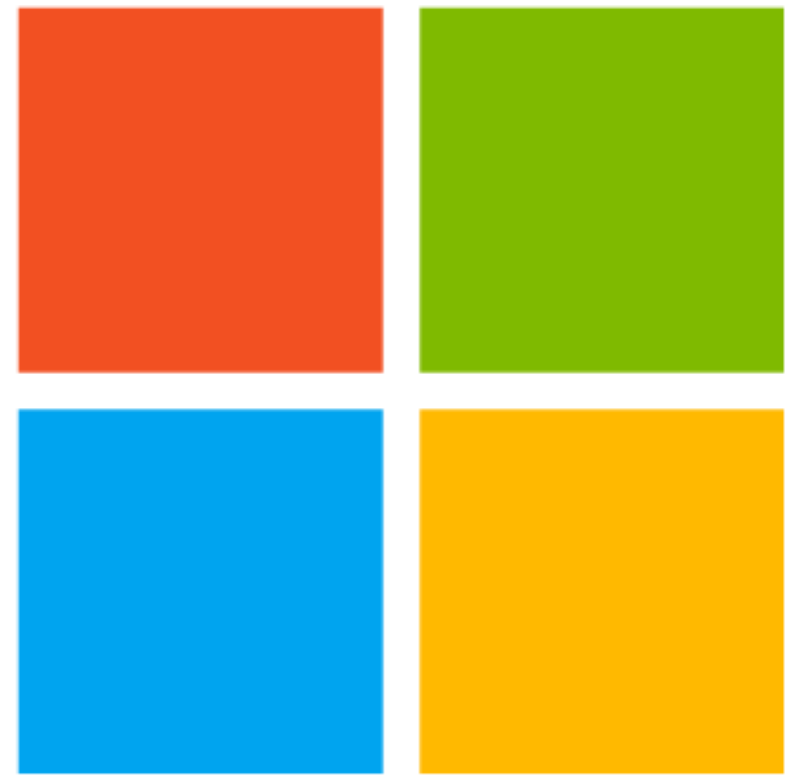


The real revolution

- Mobile phones came to Australia in 1981
- In 1991 Toyota were selling a car with a fax machine
- Useful data speeds came with 4G
- It also led to a much wider adoption of wifi.



amazon



Microsoft

Google



What's available?

Periodic Table of Amazon Web Services																				
C cloud compute																			At athena	
Ls lightsail	S simple storage																		Em emr	
Cs container service	Fs file system	Ga gamelift															Cc codecommit	Il iot 1-click	Ks kinesis video streams	Cs cloudsearch
Ks kubernetes service	Sg storage gateway	Ax alexa for business	Mi migration hub	V vpc	Q simple queue	Mn mobile hub	R relational database	Cw cloudwatch	Op opsworks	Cb codebuild	Id iot device management	Mc mediaconvert	Es elasticsearch							
λ lambda	Gl glacier	Ch chime	Ap app discovery	Cf cloudfront	No simple notificaion	Ac app sync	Ec elasticahe	As auto scaling	Sc service catalog	Cd codedeploy	Ia iot analytics	MI medialive	K kinesis							
Ba batch	Ws workspaces	Wd workdocs	Dm database migration	Rt route 53	Sf step functions	Df device farm	Ne neptune	Cf cloudformation	Sm systems manager	Cp code pipeline	Gg greengrass	Mp media package	Qs quicksight							
Eb beanstalk	Ap appstream 2.0	Wm workmail	Sm server migration	Ag api gateway	Mq mq	Ma mobile analytics	Re redshift	Ct cloudtrail	Ta trusted advisor	Cg cloud9	Fr freertos	Ms mediastore	Dp data pipeline							
Cn connect	Pn pinpoint	Em simple email	Sb snowball	Dx direct connect	Wf simple workflow	Su sumarian	D dynamodb	Co config	Ms managed services	X x-ray	Ie iot device defender	Mt mediatailor	Gu glue							
Im iam	Co cognito	Sc secrets manager	Gd guardduty	In inspector	Ma macie	So single sign-on	Cm certificate manager	Hs hsm	Ds directory	Wa waf and shield										
Sm sagemaker	Co comprehend	DI deeplens	Lx lex	ML machine learning	Po polly	Rk rekognition	Tb transcribe	Tt translate	Km key management	Ar artifact										

Case study

Resident identification

Requirements

- Convenience
- Respect dignity and privacy
- Accuracy
- These aren't in order, they are *sine qua non*
- It should be cost effective

A solution

- Facial recognition technology
 - Controversial at the moment - but not the way we are using it
- The information never leaves our control
 - It is never used for any other purpose
- This is not Clearview.ai

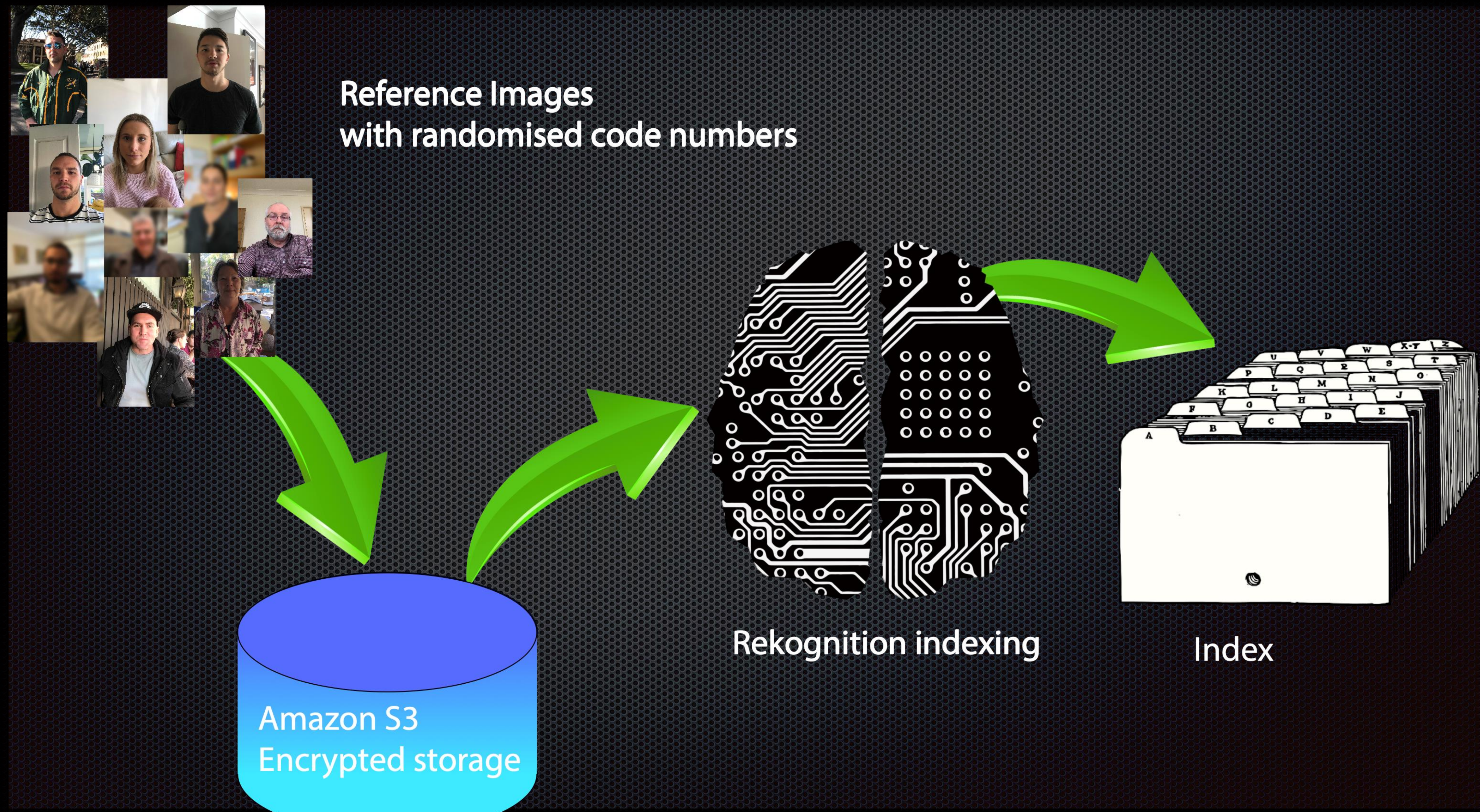
Our solution

- Amazon's facial recognition service - Rekognition
- Microsoft - Face
- Google Cloud Vision - face detection but not recognition

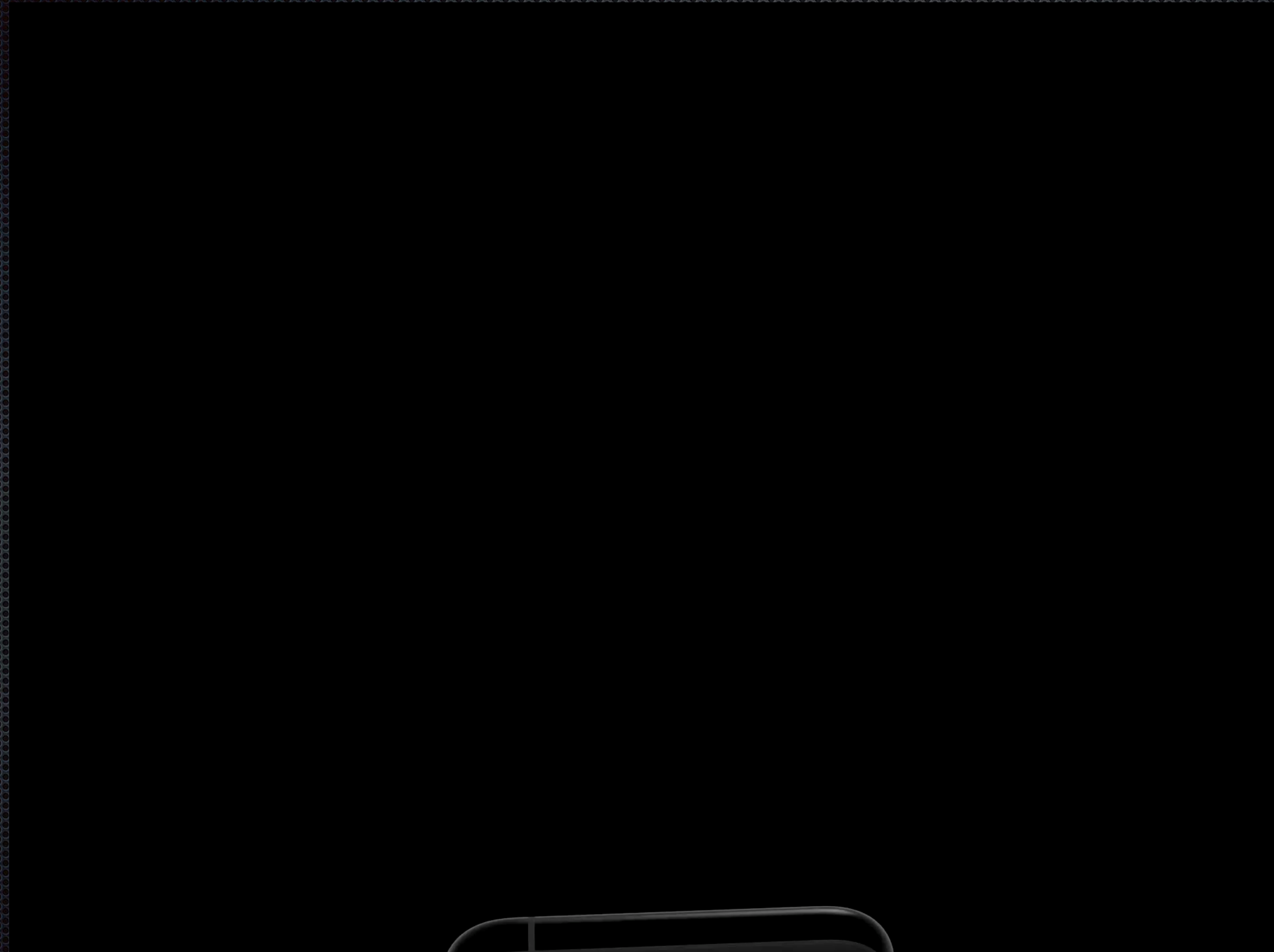
How it works

- Rekognition relies on a library of possible matches
- An “unknown” image is matched to that library
- A possible match from the library is identified, along with a confidence level

Rekognition - indexing



The code - image uploading



The code - image uploading

```
public class func saveExternalImage(image: UIImage, residentId: String, isRestricted: Bool)
{
    let filename = "\\(residentId)-(isRestricted ? "NOREF" : "REF").jpg"
    guard var url = URL(string: "https://\(Settings.string(.apiHost))") else { return }
    url.appendPathComponent(Settings.string(.apiPath))
    url.appendPathComponent("uploadImage.php")
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    let boundary = "Boundary-\\(UUID().uuidString)"
    request.setValue("multipart/form-data; boundary=\\(boundary)", forHTTPHeaderField: "Content-Type")
    guard let imageData = image.jpegData(compressionQuality: 0.5) else { return }
    let dataString = "\\(residentId)\\referenceImage"
    let body = NSMutableData()
    ... Code to set up the upload ...
    body.append(imageData)
    body.append(NSString(format: "\\r\\n--%@\\r\\n", boundary).data(using: String.Encoding.utf8.rawValue) ?? Data())
    request.httpBody = body as Data
    let task = URLSession.shared.dataTask(with: request as URLRequest) { data, response, error in
        guard error == nil else { return }
        guard let data = data else { return }
        guard let response = response as? HTTPURLResponse else { return }
        if response.statusCode == 200 {
            let encrypted = String(data: data, encoding: .utf8)
            let answer = encrypted?.decrypted(key: Settings.masterKey)
            let parts = answer?.components(separatedBy: "\\t")
            if parts?.count == 2 {
                if parts?[1] == "Success" && parts?[0] == residentId {
                    let userInfo: [String: Any] = ["ResidentID": residentId, "Result": "success"]
                    let notification = Notification(name: .ImageUploaded, object: self, userInfo: userInfo)
                    NotificationCenter.default.post(notification)
                } else {
                    let userInfo: [String: Any] = ["ResidentID": residentId, "Result": "failure"]
                    let notification = Notification(name: .ImageUploaded, object: self, userInfo: userInfo)
                    NotificationCenter.default.post(notification)
                }
            }
        }
    }
    task.resume()
}
```


The code - image uploading

```
$decrypted = uCareDecrypt($encrypted, $key);
$parts = explode("\t", $decrypted);
$returnData = "";
if (count($parts) == 2) {
    $photoIdentifier = $parts[0];
    $photoType = $parts[1];
    $target_dir = "../secure/tempDir";
    if (file_exists($target_dir)) {
        $fromName = $_FILES["theImage"]["tmp_name"];
        $toName = $_FILES["theImage"]["name"];
        $destination = $target_dir . "/" . $toName;
        move_uploaded_file($fromName, $destination);
        if (file_exists($destination)) {
            transferFileToS3($destination, $facilityId, $photoType);
            if ($photoType == "referenceImage") {
                $imageId = substr($photoIdentifier, 0, 40);
                $returnData = "{$imageId}\t";
                reindexFaces($facilityId, $destination, $imageId);
            } else if ($photoType == "messageImage") {
                $imageId = substr($photoIdentifier, 0, 40);
                $returnData = "{$imageId}\t";
            }
        }
    }
    $returnData = $returnData . "Success";
} else {
    $returnData = $returnData . "Failure";
}
$data = uCareEncrypt($returnData, $key);
echo $data;
}
```


The code - image indexing

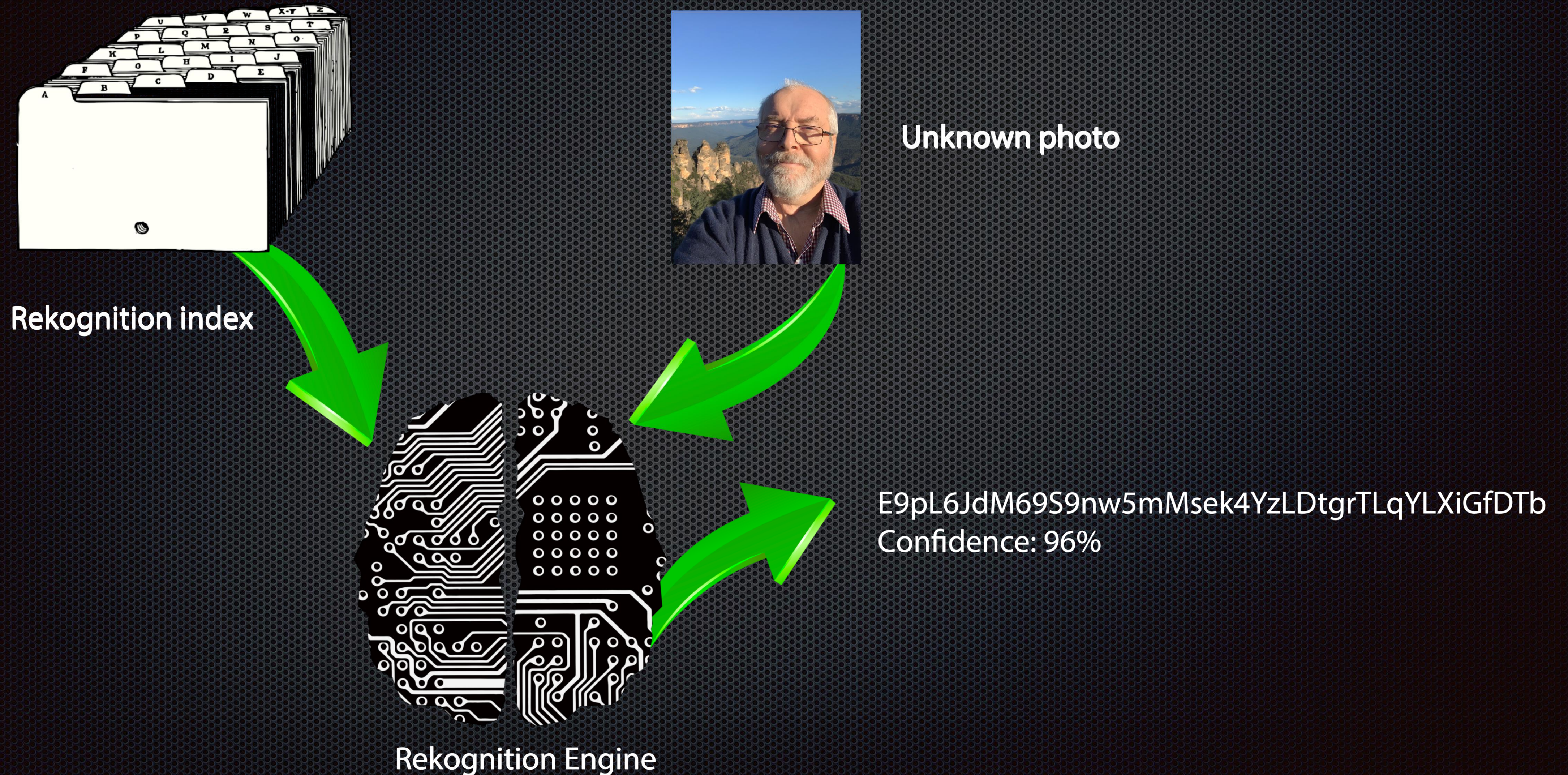
```
function reindexFaces($facilityCode, $imageName, $residentId) {
    $collectionName = $facilityCode . "_Collection";
    $awsKey = getAWSKey();
    $awsSecret = getAWSSecret();
    $rekognitionClient = new RekognitionClient(['version' => 'latest', 'region' => 'ap-southeast-2', 'credentials' => ['key' => $awsKey, 'secret' => $awsSecret]]);

    // The following code will remove the collection - it is for testing purposes only.
    // The whole block should be commented out when in production.
    //
    //$result = $rekognitionClient->deleteCollection(['CollectionId' => $collectionName]);

    // Does the collection exist?

    try {
        $result = $rekognitionClient->describeCollection(['CollectionId' => $collectionName]);
        $statusCode = $result["@metadata"]["statusCode"] . PHP_EOL;
    }
    catch (Exception $e) {
        $result = $rekognitionClient->createCollection(['CollectionId' => $collectionName]);
    }
    $basename = basename($imageName);
    $keyName = "{$facilityCode}/ReferenceImages/{$basename}";
    $result = $rekognitionClient->indexFaces(['CollectionId' => $collectionName,
        'DetectionAttributes' => ['DEFAULT'],
        'ExternalImageId' => $residentId,
        'Image' => ['S3Object' => ['Bucket' => 'ucarelive-s3', 'Name' => $keyName]],
        'MaxFaces' => 1,
        'QualityFilter' => 'AUTO']);
}
```


Rekognition - identifying



The code - identification



The code - identification

```
$result = uCareDecrypt($security, $key);
if ($result == "Secure") {
    $target_dir = "../../secure/tempDir";
    if (file_exists($target_dir)) {
        $fromName = $_FILES["messageImage"]["tmp_name"];
        $data = file_get_contents($fromName);
        $collectionName = $facilityCode . "_Collection";
        $awsKey = getAWSKey();
        $awsSecret = getAWSSecret();
        $rekognitionClient = new RekognitionClient(['version' => 'latest', 'region' => 'ap-southeast-2', 'credentials' => ['key' => $awsKey, 'secret' => $awsSecret]]);
        $result = $rekognitionClient->searchFacesByImage(['CollectionId' => $collectionName, 'Image' => ['Bytes' => $data]]);
        $imageId = $result['FaceMatches'][0]['Face']['ExternalImageId'];
        $confidence = $result['FaceMatches'][0]['Face']['Confidence'];

        $returnData = $imageId . "\t" . $confidence;
    } else {
        $returnData = "Failure";
    }
}
echo uCareEncrypt($returnData, $key);
}
```


The code - presenting the result

```
func imageRecognised(result: (resident: Resident, confidence: Float)) {
    DispatchQueue.main.async {
        let response = ""
        I have identified this person as \(result.resident.name) with \(result.confidence)% confidence.

        Machine identification is not yet perfect, so please verify this person's identity by another means before relying on it for medical or other purposes."
        ""

        self.resultLabel.backgroundColor = UCTheme.placeholderColour
        self.resultLabel.textInsets = UIEdgeInsets(top: 8.0, left: 16.0, bottom: 8.0, right: 16.0)
        self.resultLabel.labelText = response
        self.resident = result.resident
    }
}
```


- Margaret Hamilton was awarded the Presidential Medal of Freedom in 2016



- David Reidy
 - Founder and lead developer for uCare
- david@uca.re
- @reidyd
- Stand 08 in the exhibitor area

Full code listings


```

public class func saveExternallImage(image: UIImage, residentId: String, isRestricted: Bool)
{
    let filename = "\(residentId)-\(isRestricted ? "NOREF" : "REF").jpg"
    guard var url = URL(string: "https://\(Settings.string(.apiHost))") else { return }
    url.appendPathComponent(Settings.string(.apiPath))
    url.appendPathComponent("uploadImage.php")
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    let boundary = "Boundary-\(UUID().uuidString)"
    request.setValue("multipart/form-data; boundary=\(boundary)", forHTTPHeaderField: "Content-Type")
    guard let imageData = image.jpegData(compressionQuality: 0.5) else { return }
    let dataString = "\(residentId)\treferenceImage"
    let body = NSMutableData()
    body.append(NSString(format: "\r\n--%@\r\n", boundary).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "Content-Disposition: form-data; name=\"userId\"\r\n\r\n" as NSString).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: (Settings.string(.userId) as NSString)).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "\r\n--%@\r\n", boundary).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "Content-Disposition: form-data; name=\"data\"\r\n\r\n" as NSString).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: (dataString as NSString)).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "\r\n--%@\r\n", boundary).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "Content-Disposition: form-data; name=\"theImage\"; filename=\"\(filename)\"\r\n" as NSString).data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(NSString(format: "Content-Type: image/jpg\r\n\r\n").data(using: String.Encoding.utf8.rawValue) ?? Data())
    body.append(imageData)
    body.append(NSString(format: "\r\n--%@\r\n", boundary).data(using: String.Encoding.utf8.rawValue) ?? Data())
    request.httpBody = body as Data
    let task = URLSession.shared.dataTask(with: request as URLRequest) { data, response, error in
        guard error == nil else { return }
        guard let data = data else { return }
        guard let response = response as? HTTPURLResponse else { return }
        if response.statusCode == 200 {
            let encrypted = String(data: data, encoding: .utf8)
            let answer = encrypted?.decrypted(key: Settings.masterKey)
            let parts = answer?.components(separatedBy: "\t")
            if parts?.count == 2 {
                if parts?[1] == "Success" && parts?[0] == residentId {
                    let userInfo: [String: Any] = ["ResidentID": residentId, "Result": "success"]
                    let notification = Notification(name: .ImageUploaded, object: self, userInfo: userInfo)
                    NotificationCenter.default.post(notification)
                } else {
                    let userInfo: [String: Any] = ["ResidentID": residentId, "Result": "failure"]
                    let notification = Notification(name: .ImageUploaded, object: self, userInfo: userInfo)
                    NotificationCenter.default.post(notification)
                }
            }
        }
    }
    task.resume()
}

```