Jahangirnagar University

Institute of Information Technology

An Optimized RESTful E-Governance Application Framework for People Identity Verification in Cloud

Presented By:
Ahmedur Rahman Shovon
Roll: 160029

Supervised By:
Dr. Md. Whaiduzzaman,
Associate Professor,
Institute of Information Technology,
Jahangirnagar University

Problem Statement (1/2):

-> People identity verification using face recognition, police clearance and information validation is a long process.

-> This delay happens due to the direct involvement of people and inaccessibility towards government maintained citizen database.

-> Faces of individuals are increasing day by day but are not properly mapped.

-> As a result, Government can not track the change of face identities of the individuals.

Problem Statement cont’d (2/2):

-> To verify an individual’s identity, an organization needs to perform:

- Vital information validation from Government database (no public access)
- Face recognition of current photo and authorized photo (no face recognition system provided by Government)
- Clearance of police records of that individual (manual process that requires local police involvement)

-> Digitization of identity verification is a very promising sector to work on.

Related Works (1/3):

Real time face recognition system (RTFRS) [1]:

- Developed a Windows based desktop application for face recognition.
- Error rate under normal conditions: 3%.

A study of face recognition as people age [2] [21] [22]:

- Studied face recognition across ages within passport photo verification task.
- Showed how age differences affect recognition performance.
- Observed that difficulty of face recognition algorithms saturated after the age gap is larger than four years (up to ten years).

Related Works cont’d (2/3):

Face Recognition: A Literature Survey [3], [25]:

- Provided an up-to-date critical survey of face recognition research.
- Showed insights of machine recognition of faces.
- Listed typical applications of face recognition in public services.

Using cloud computing for e-government: challenges and benefits [4]:

- Discussed about the benefits of using cloud computation in E-governance.
- Presented government’s enhanced ability to interact and collaborate.

Related Works cont'd (3/3):

Cloud Based E-governance Services [5], [6], [7], [23]:

- Showed cloud computing advantages in various parts of E-government [20].
- Availability, cost efficiency, scalability, storage capacity and security are enlisted.
- Proposed a cloud based model for national E-governance plan [8].

RESTful Application Framework [9], [24]:

- RESTful based web technology is gaining attractions in developed countries.
- Integrated RESTful web services and cloud computing [10].
- Proposed a model to design and describe REST API maintaining constraints [11].
- Demonstrated good practice of developing RESTful application framework [12].

Objectives (1/2):

- Survey existing people identity verification systems.
- Identify need of RESTful approach for people identity verification.
- Analyze pros and cons of cloud based solution to develop the framework for a successful E-governance system.
- Compare cloud based face recognition services.

Objectives cont’d (2/2):

- Present an algorithm for people identity verification using face recognition, police clearance and information validation.
- Develop an optimized RESTful E-Governance Application Framework based on the algorithm.
- Evaluate the framework’s accuracy, scalability and reliability based on resource utilization, application performance, and operational health.

Existing Research Limitations:

- No cloud based people identity verification framework is proposed.
- Face recognition is not mapped with people’s database.
- Authorized organizations can not access government maintained citizen database.
- No RESTful API is developed to automate people identity verification.
- No deep learning-based image recognition services is used.
- No deployment model is shown to search, verify and organize millions of images in cloud.

Overview:

- An optimized people identity verification framework.
- Utilization of cloud for E-Governance application.
- RESTful approach for better and secure service provision.
- Deep learning based face recognition using cloud services mapped with individuals.

Block Diagram:

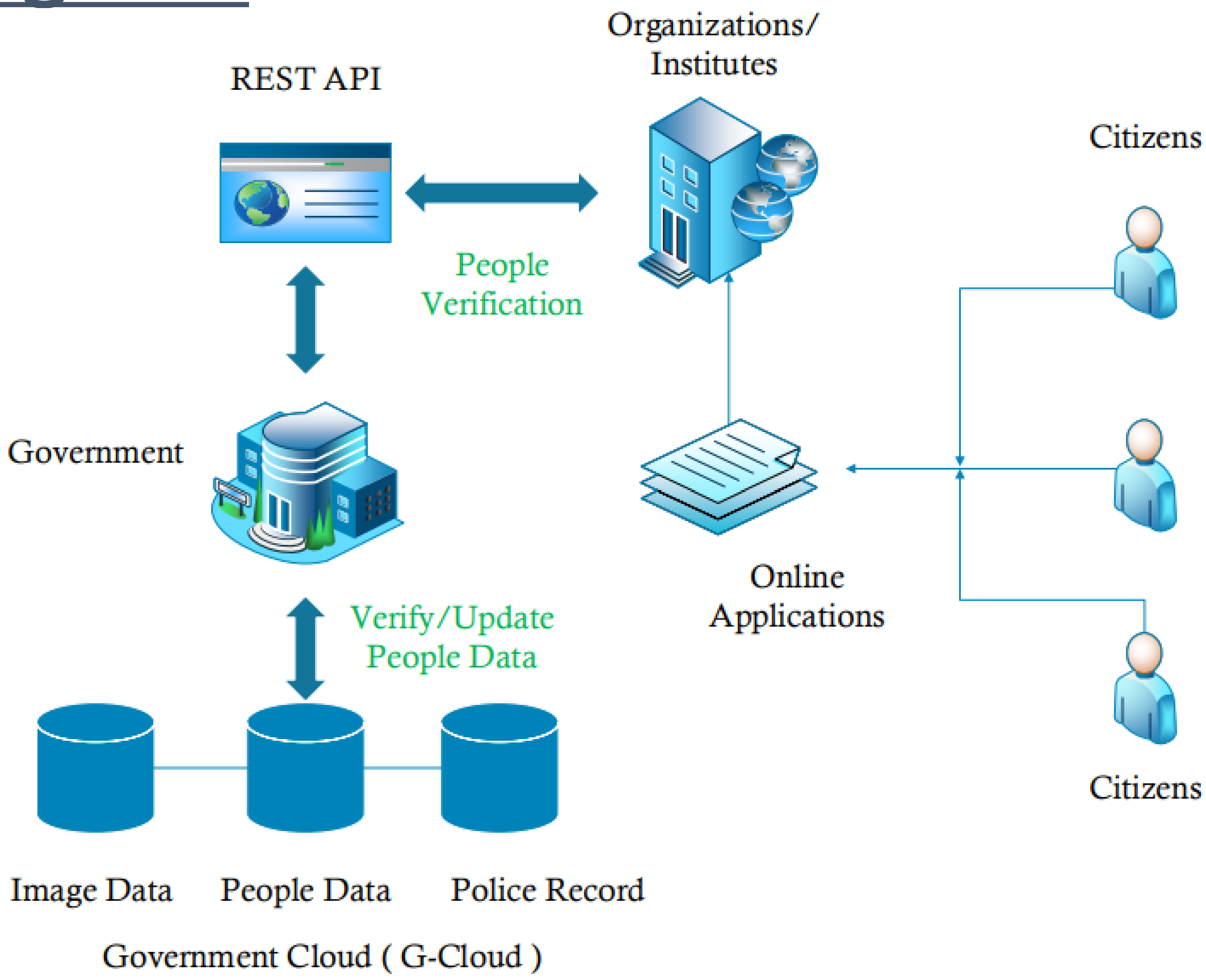

Figure: Block Diagram of People Identity Verification Application Framework

Application Framework Architecture (1/6):

- Framework has two parts:
  - REST API
  - API Dashboard
- Developed using Python Flask Framework.
- Uses Amazon Web Services (AWS) [13].
- JSON Web Token (JWT) [14] provides JSON based access tokens.
- Deep learning based AWS Rekognition services [15] for face recognition and facial vector data storage.
- SQLAlchemy, an ORM is connected with MySQL database [16].
- ELB and Auto Scaling ensures scalability [17].
- CloudWatch monitors AWS resources [18].

Architecture cont'd – System Architecture (2/6):

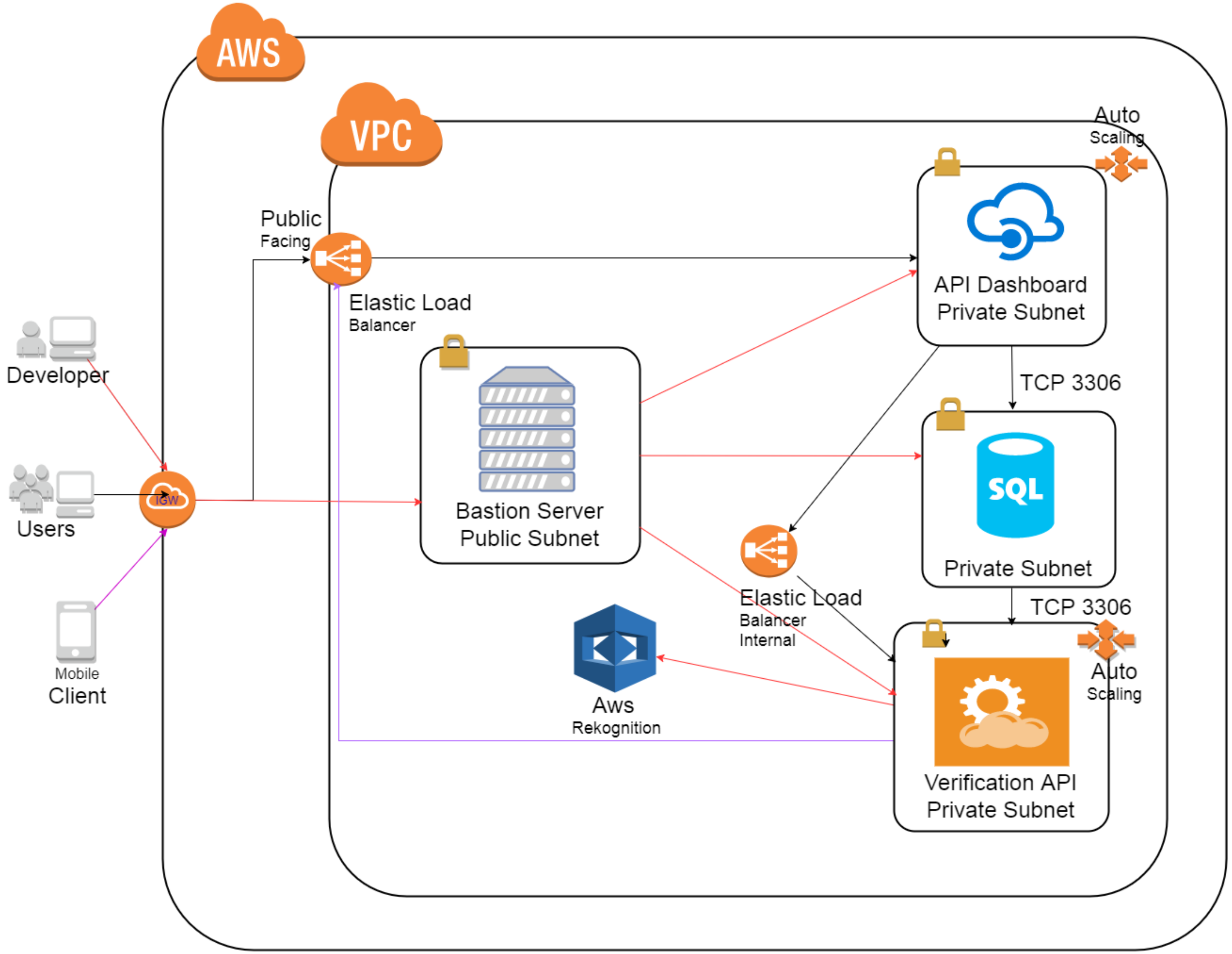

Figure: System Architecture of People Identity Verification Application Framework

Architecture cont'd – Software (3/6):

Software	Description
Python	Version 3.6.1
Flask	Version 0.12.2
MySQL	Version 5.7.19
SQLAlchemy	SQLAlchemy version 1.1.12 Flask-SQLAlchemy version 2.2
JSON Web Tokens	PyJWT version 1.5.2 Flask-JWT version 0.3.2
AWS CLI	AWSCLI version 1.11.112 Boto3 version 1.4.4 Boto-core version 1.5.75

Table: Software Description of People Identity Verification Application Framework

Architecture cont’d – Hardware (4/6):

Hardware	Description
VPC	3 private subnets and 1 public subnet
Bastion Node	Access the others servers as it includes the public inbound rules and private outbound rules
API & Dashboard Node	Flask API and dashboard node. All AWS services like Rekognition, AMI, AWSCLI are integrated.
Database Node	Database is hosted in this node. Configured with only private access from other nodes.
Load Balancers	2 Elastic Load Balancers handle all inbound traffic
Security Groups	Handles inbound and outbound traffic in each node

Table: Hardware Description of People Identity Verification Application Framework

Architecture cont'd – API Design (5/6):

- **Stateless Design:** No session storage
- **Self-descriptive Messages:** Simple request-response format
- **Semantics:** Use features of the HTTP protocol including
  - HTTP Verbs
  - HTTP Status Codes
  - HTTP Authentication
- **I/O Format:** JSON
- **URL Structure:** Descriptive, utilized natural hierarchy of path structure
- **Authentication:** JWT based access token generated by credentials
- **Timestamps:** ISO-8601 standard [14]
- **Error Handling:** Returns semantic HTTP status code in each response

Architecture cont'd – Database Design (6/6):

Figure: Database Model Diagram of People Identity Verification Application Framework

Process Flowchart:

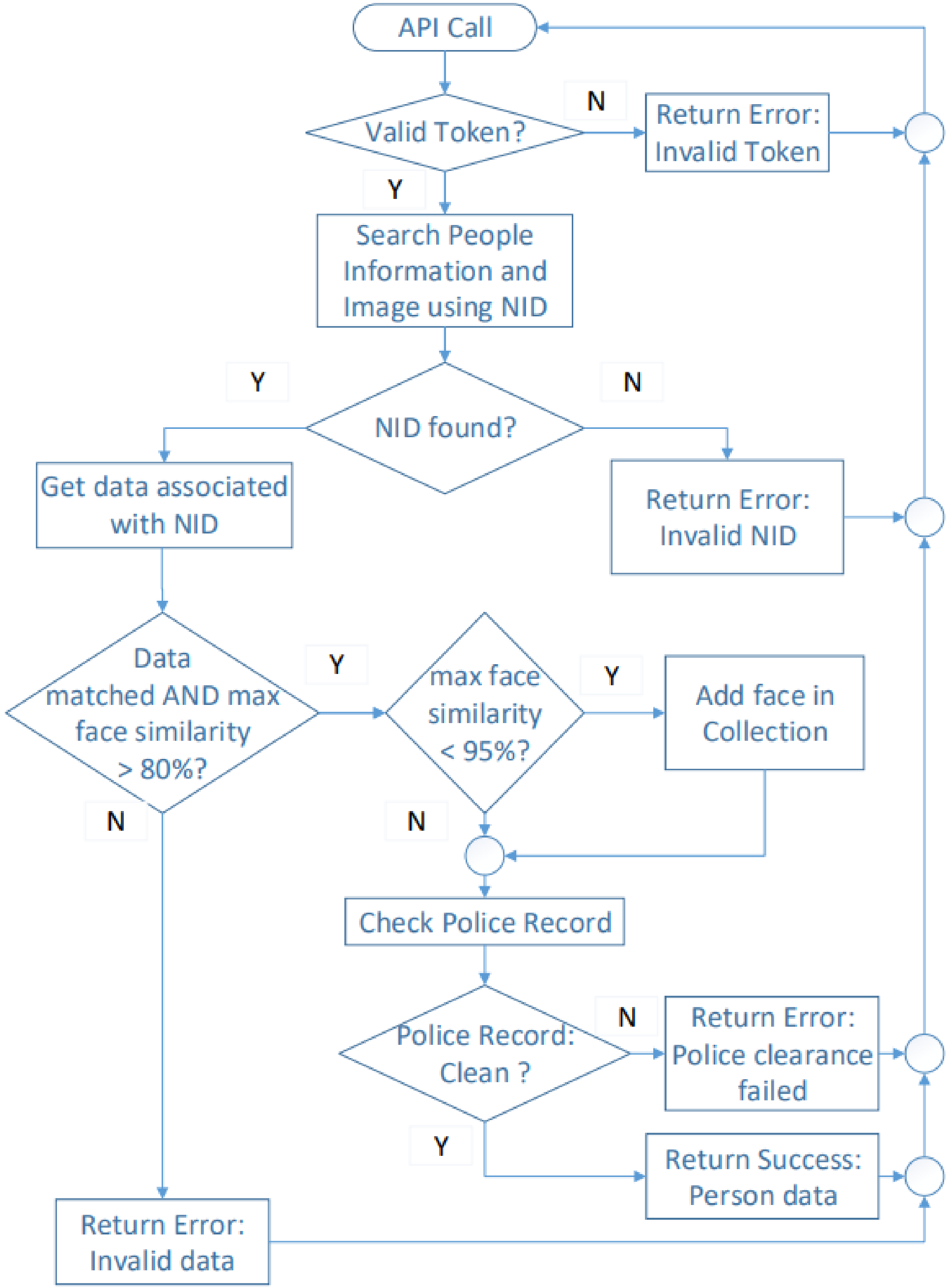

Figure: Process flowchart

Algorithm:

```
API call;
if TOKEN in Request = True then
    if Validity_TOKEN = True then
        if NID, personal information & image in Request then
            if NID_found = True then
                if personal information & image matches with database then
                    Get Face_List from Database;
                    Set Max_Score = 0 ;
                    foreach Face in Face_List do
                        Calculate similarity score;
                        Set Max_Score = max(score, max_score);
                    end
                    if Max_Score > 80 then
                        if Max_Score < 95 then
                            Store image in Collection;
                        end
                        Get Records_Police ;
                        if Clearence = True then
                            Return: Verification Success with all data;
                        end
                        else
                            Return Error with case record details;
                        end
                    end
                    else
                        Return Error: Image match failed;
                    end
                end
                else
                    Return Verified = True;
                end
            end
            else
                Return Error: NID not found;
            end
        end
        else
            Return Error: Missing mandatory fields;
        end
    end
    else
        Return: TOKEN not valid;
    end
end
else
    Return: TOKEN not found;
end
```

Evaluation – Face Recognition Accuracy (1/6):

- **Constraint:** Primarily Face collection contains one face per person
- **Test Face Set:** Face Recognition Technology(FERET) database [19]
- **First Iteration Result:** Error rate = 1.60% (one face per person) (better than [1])
- **Second Iteration Result:** Error rate = 1.13% (avg. two faces per person)

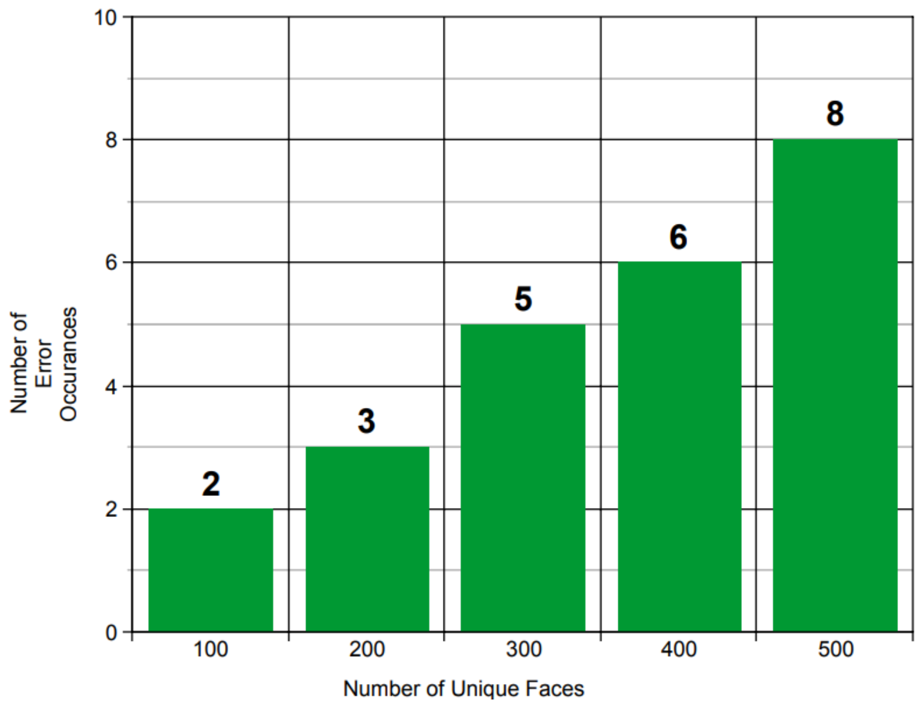

Figure: First Iteration Error Chart

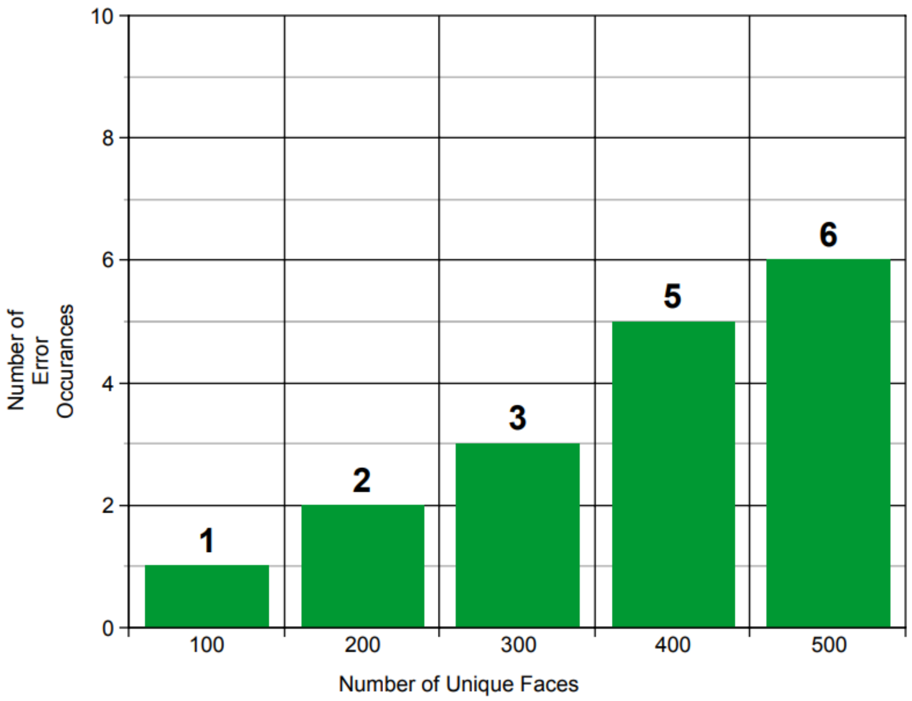

Figure: Second Iteration Error Chart

Evaluation cont'd – Load Testing (2/6):

- **Metric:** Network packet counts (CloudWatch Metric)
- **Methodology:** Custom script generates random number of hits for a time period
- **Axis Information:** X axis – time frame**,** Y axis – number of network packets
- **Result:** Normal operational health. (Better than non cloud hosts)

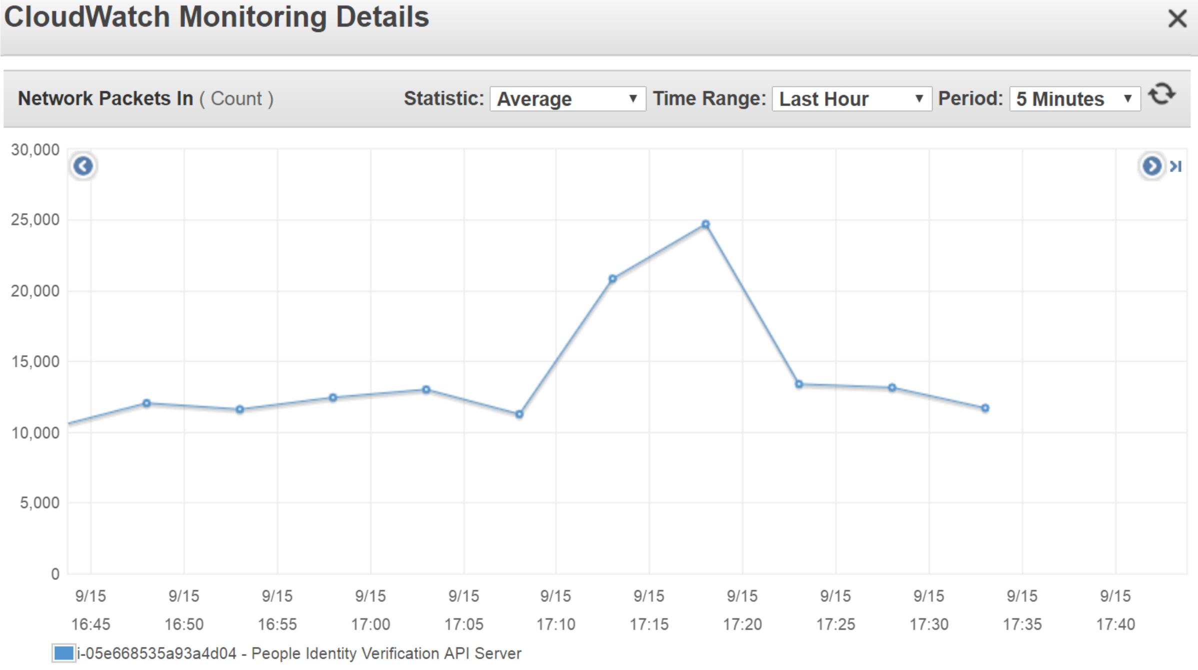

Figure: API Packet Counts

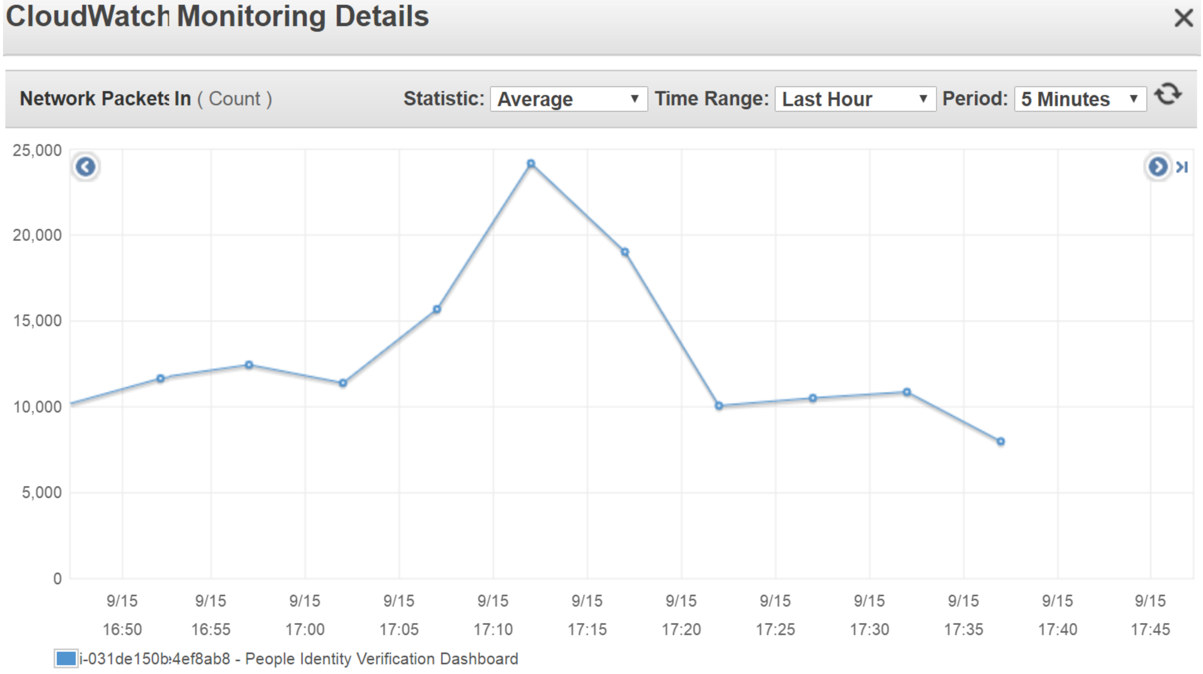

Figure: Dashboard Packet Counts

Evaluation cont'd – Scalability (3/6):

- **Metric:** CPU Utilization (CloudWatch Metric)
- **Methodology:** Custom script consumes arbitrary CPU power for a time period
- **Axis Information:** X axis – time frame**,** Y axis – CPU utilization
- **Result:** Normal operational health. No system damage. (Reliable than static hosts)

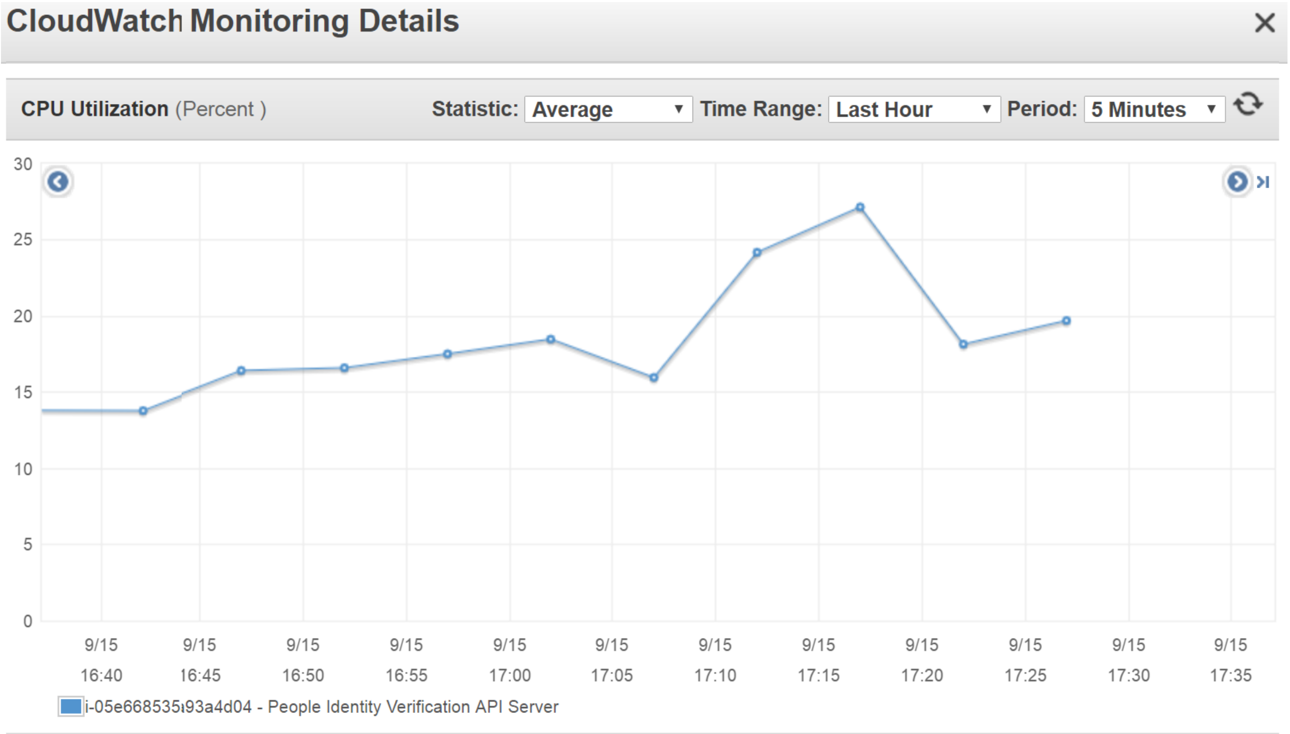

Figure: API Node CPU Utilization

Evaluation cont'd – Availability (4/6):

- **Metric:** Auto Scaling (CloudWatch Metric)
- **Methodology:** Custom script consumes arbitrary CPU power for a time period
- **Axis Information:** X axis – time frame, Y axis – CPU utilization
- **Result:** Generated alert for low and excessive CPU utilization added and detached additional computing resources as needed (Better than client – server hosting)

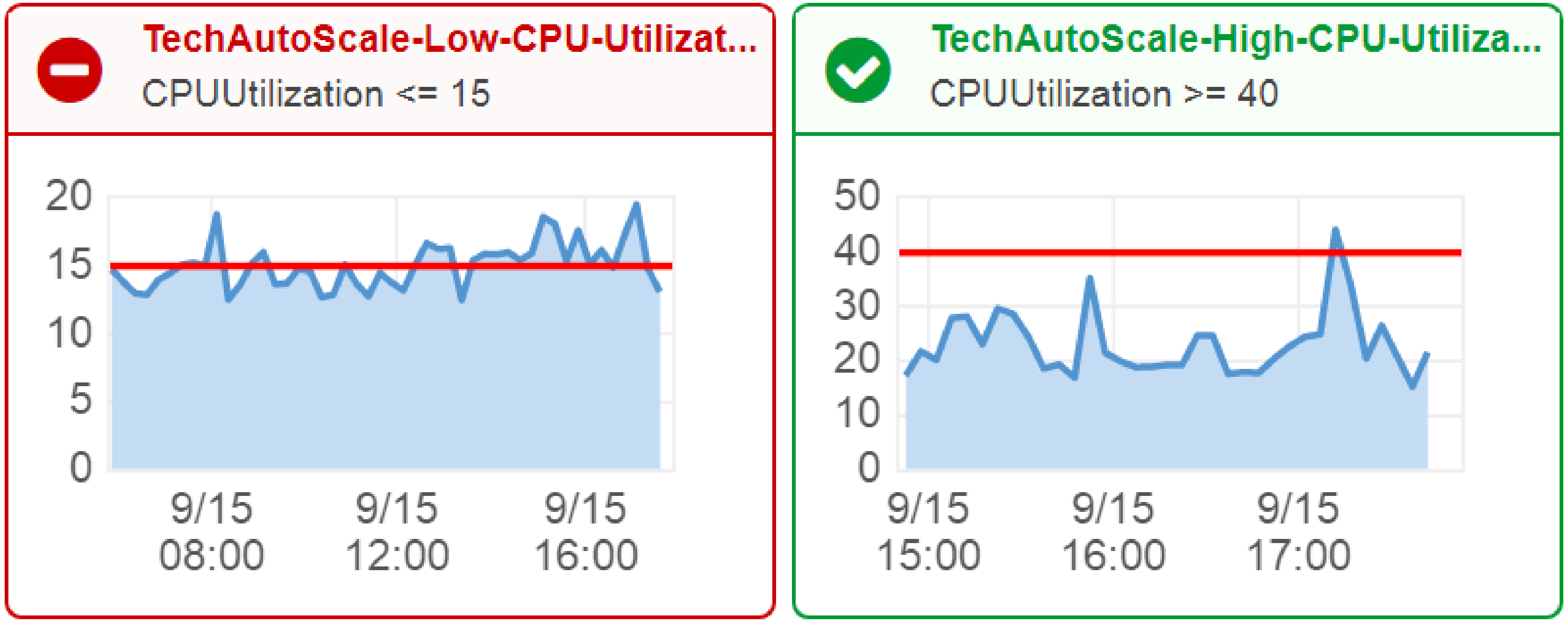

Figure: Generated Alert when the CPU utilization was under 15% for 8 hours

Evaluation cont’d – API HTTP Status Code (5/6):

- **Metric:** Elastic Load Balancer HTTP Status Code Counts (CloudWatch Metric)
- **Methodology:** Log HTTP status code in ELB for a time period
- **Axis Information:** X axis – time frame**,** Y axis – status count
- **Result:** 99.65% Health Hosts Count (API integration is proved successful)

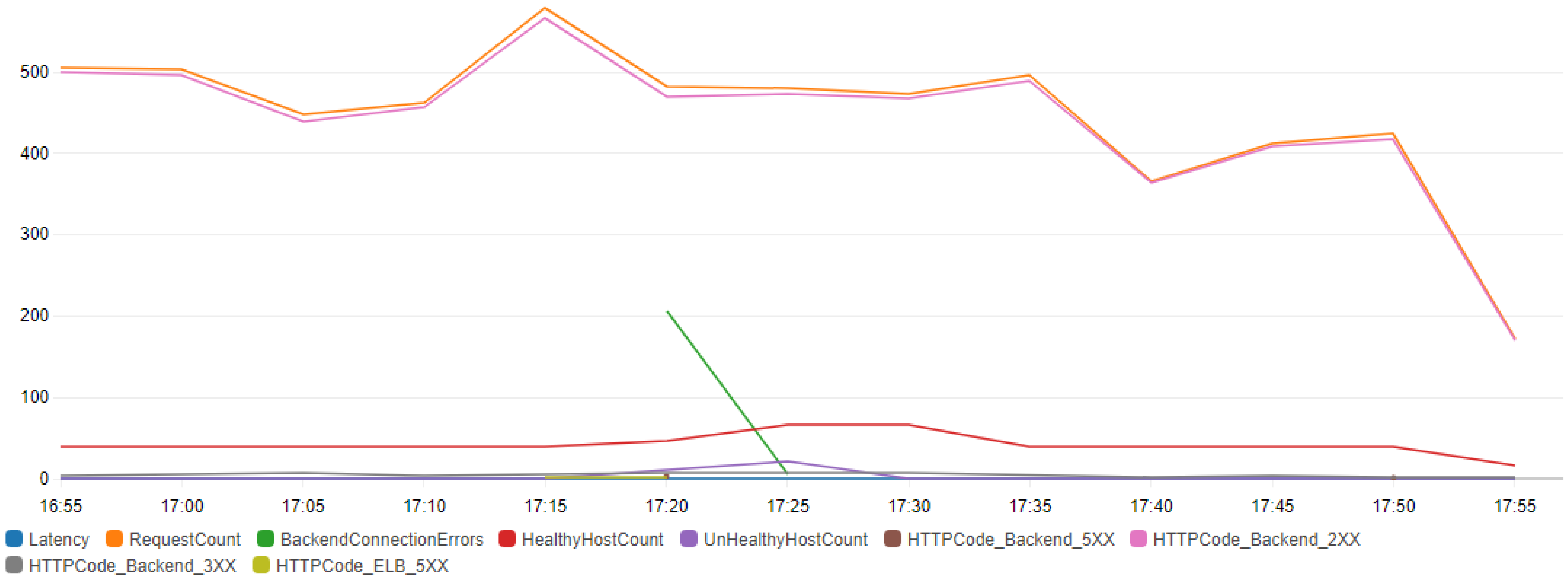

Figure: Elastic Load Balancer HTTP Status Code Counts

Evaluation cont'd – Cost Efficiency (6/6):

- **Metric:** AWS Services Cost Metrics (CloudWatch Metric)
- **Methodology:** Log service costs within a date range
- **Axis Information:** X axis – date, Y axis – costs by different services
- **Result:** Dynamic costing based on consumed services.

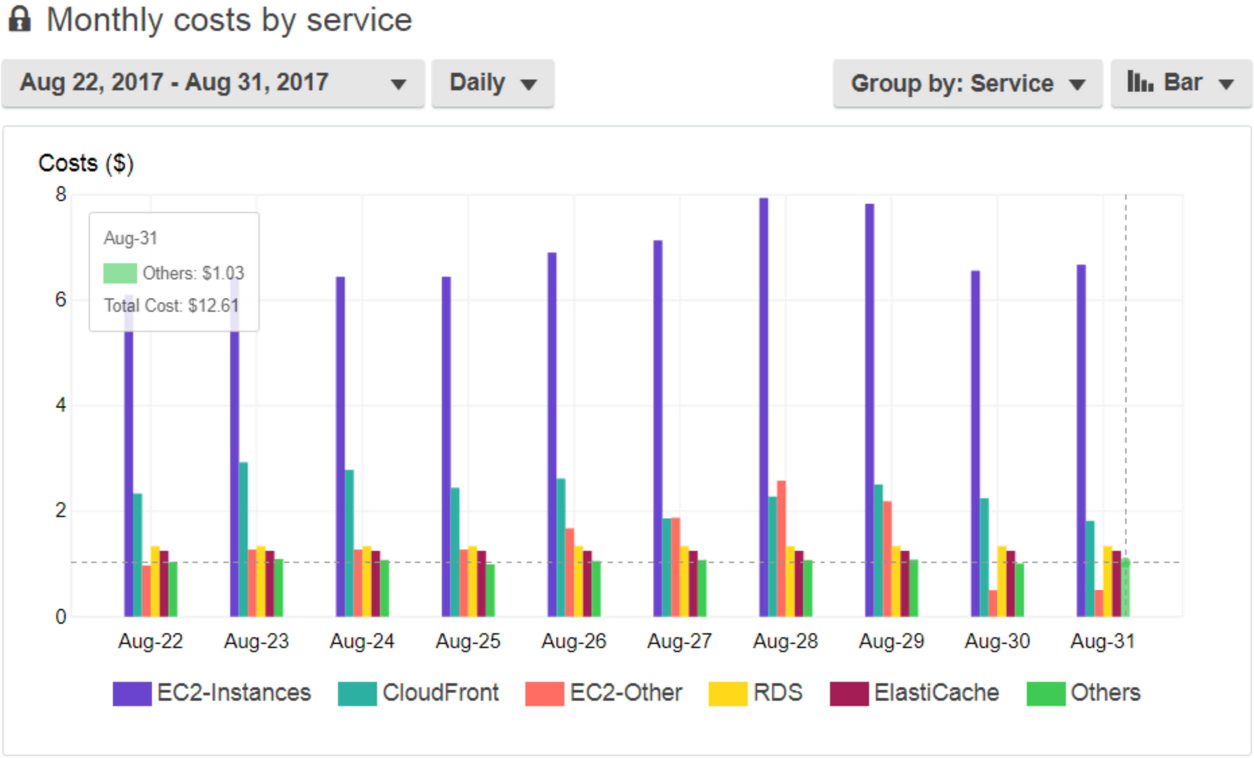

Figure: Monthly costs by consumed services

Result – Request Sample (1/3):

- **Authorization:** Any organization with valid access token can send request
- **API Path:** Request should follow fixed API path
- **Mandatory Fields:** 6 mandatory fields should be attached with each request
- **Platform / Framework Dependency:** No platform / framework dependency

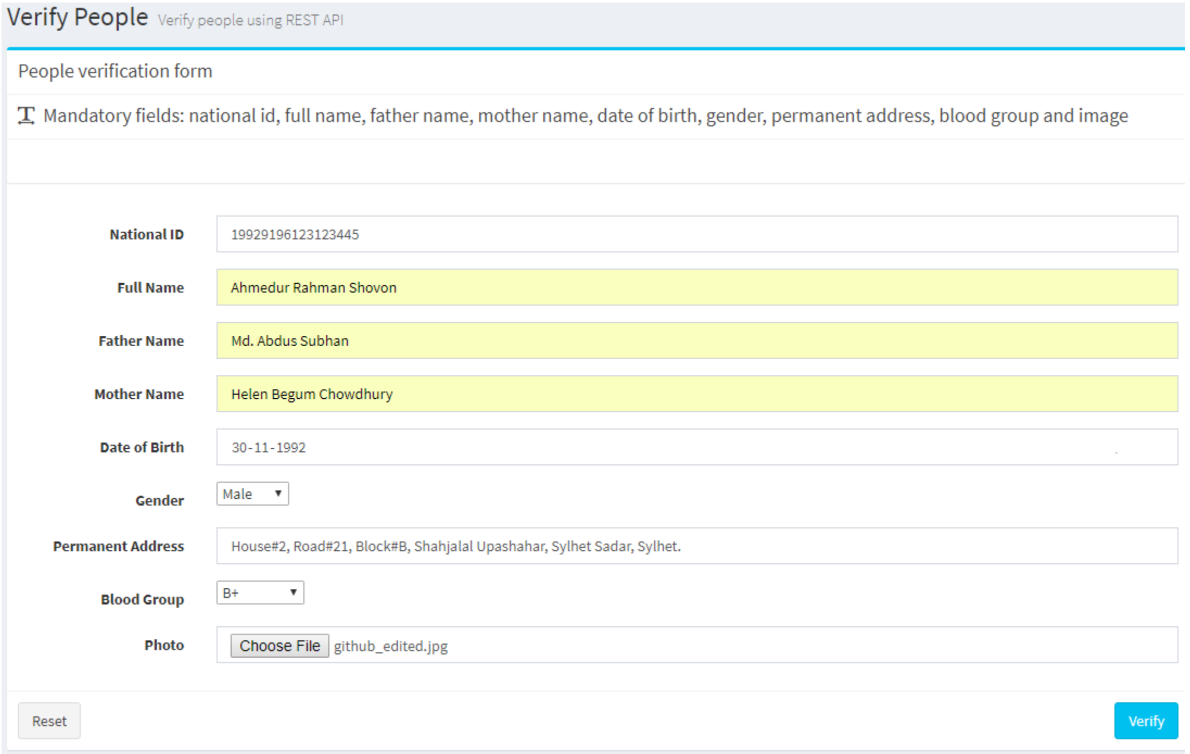

Figure: Input form sample for People Identity Verification Application Framework

Result – Response (2/3):


```
object{
  "status_code":string"200",
  "status":string"valid",
  "data":object{
    "personal_information":object{
      "national_id":string"19929196123123445",
      "full_name":string"Ahmedur Rahman Shovon",
      "father_name":string"Md. Abdus Subhan",
      "mother_name":string"Helen Begum Chowdhury",
      "gender":string"male",
      "address":string"House#2, Road#21, Block#B, Shahjalal Upashahar, Sylhet Sadar, Sylhet.",
      "dob":string"1992-11-30",
      "email":string"shovon.sylhet@gmail.com",
      "phone":string"+8801731246426",
      "blood_group":string"B+",
      "photo_url":string"19929196123123445_1502852020506_github_edited.jpg",
      "religion":string"Islam",
      "nationality":string"Bangladeshi"
    },
    "face_index":array[
      object{
        "face_id":string"e340bc25-feb5-5917-ade1-d91845602bb0",
        "face_timestamp":string"2017-08-16 02:53:40.504936",
        "face_url":string"29_1502852020506_github_edited.jpg"
      },
      object{
        "face_id":string"45825738-0784-5738-9978-92f3e4c3e478",
        "face_timestamp":string"2017-08-28 04:39:35.180052",
        "face_url":string"29_1503895159346_cover2.jpg"
      }
    ],
    "police_record":array[
    ]
  },
  "maximum_face_similarity":string"87%",
  "inserted_in_face_collection":string"True",
  "error":array[
  ]
}
```


Figure: Sample response from People Identity Verification Application Framework

Result – API Dashboard (3/3):

- **Authorization:** Only the Government can access API dashboard

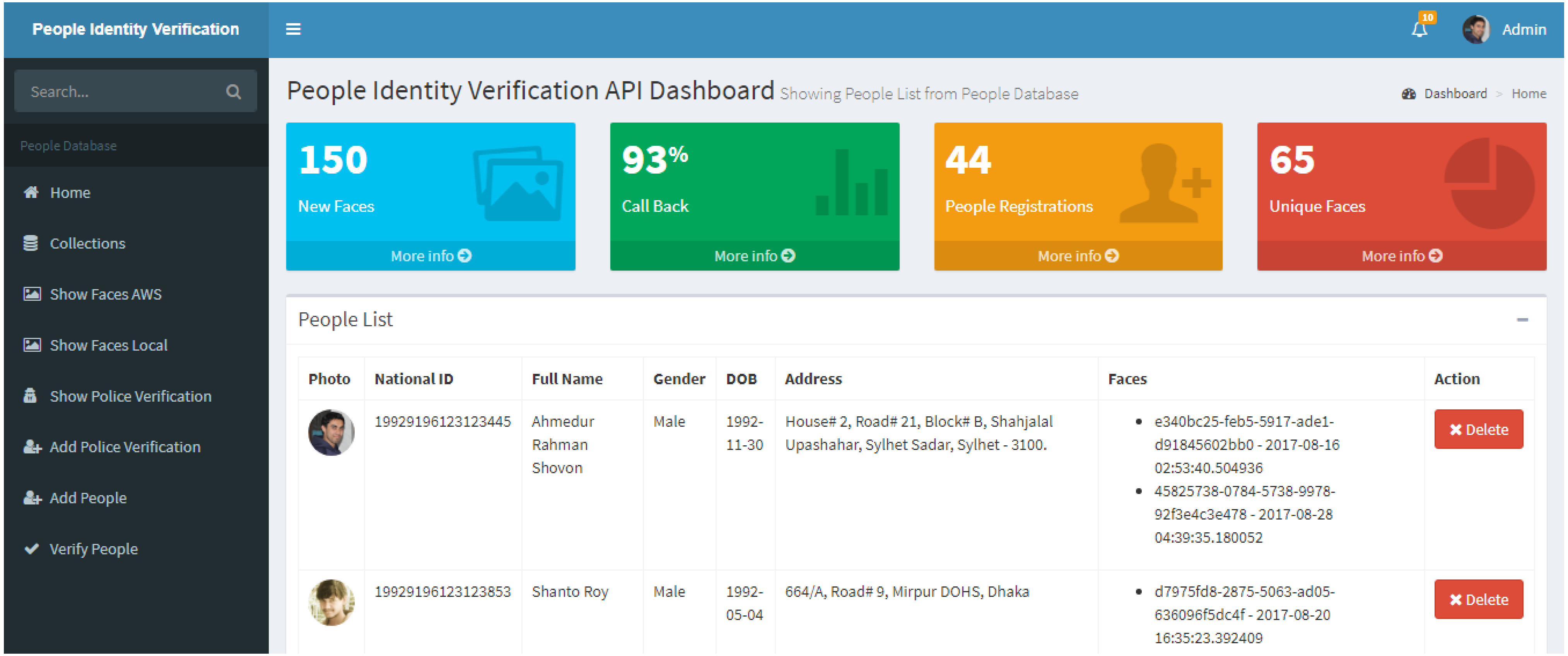

Figure: API Dashboard of People Identity Verification Application Framework

Assumptions and Limitations:

- People database should have at least one valid photo for each enlisted people.
- Initially when the database has a single image per person that may lead to a minor error in face recognition operations.
- Authenticated access tokens should be kept secret by the organizations.

Future work:

- Develop machine learning based analytics on stored face meta data.
- Identify people from Closed-Circuit Television (CCTV) footages or videos.
- Detect listed criminals from live video feeds.
- Generate alerts and inform security reinforcements after detecting criminals.
- Identify people after severe accident on transportation.
- Block unwanted access to security parameters using automated entry system.

Conclusion:

- Assured delivery of public services in a faster way eradicating long analog verification processes.
- Smoothened the path to establish better E-governance using People Identity Verification Application Framework.
- Showed improved accuracy.
- Performed benchmarking based on major factors.
- Created opportunity to utilize people face meta data for further research.

Reference:

[1] S. Haji and A. Varol. “Real time face recognition system (RTFRS)”. In: 2016 4th International Symposium on Digital Forensic and Security (ISDFS). 2016, pp. 107–111. doi: 10.1109/ISDFS.2016.7473527.

[2] Haibin Ling et al. “A study of face recognition as people age”. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE. 2007, pp. 1–8.

[3] W. Zhao et al. “Face Recognition: A Literature Survey”. In: ACM Comput. Surv. 35.4 (Dec. 2003), pp. 399–458. issn: 0360-0300. doi: 10.1145/ 954339.954342. url: http://doi.acm.org/10.1145/954339.954342.

[4] Sajjad Hashemi, Khalil Monfaredi, and Mohammad Masdari. “Using cloud computing for e-government: challenges and benefits”. In: International Journal of Computer, Information, Systems and Control Engineering 7.9 (2013), pp. 596–603.

[5] Rama Krushna Das, Sachidananda Patnaik, and Ajita Kumar Misro. “Adoption of Cloud Computing in e-Governance”. In: (2011). Ed. by Natarajan Meghanathan, Brajesh Kumar Kaushik, and Dhinaharan Nagamalai, pp. 161–172. doi: 10.1007/978-3-642-17881-8_16. url: https: //doi.org/10.1007/978-3-642-17881-8_16.

[6] R. H. Di et al. “Research on the impact of cloud computing trend on Egovernment framework”. In: 2011 International Conference on E-Business and E-Government (ICEE). 2011, pp. 1–4. doi: 10.1109/ICEBEG.2011. 5886913.

[7] D. G. Chandra and R. S. Bhadoria. “Cloud Computing Model for National E-governance Plan (NeGP)”. In: 2012 Fourth International Conference on Computational Intelligence and Communication Networks. 2012, pp. 520–524. doi: 10.1109/CICN.2012.61.

Reference(Cont’d):

[8] D. G. Chandra and R. S. Bhadoria. “Cloud Computing Model for National E-governance Plan (NeGP)”. In: 2012 Fourth International Conference on Computational Intelligence and Communication Networks. 2012, pp. 520–524. doi: 10.1109/CICN.2012.61.

[9] Wojciech Cellary and Sergiusz Strykowski. “E-government based on cloud computing and service-oriented architecture”. In: Proceedings of the 3rd international conference on Theory and practice of electronic governance. ACM. 2009, pp. 5–10.

[10] Jason H Christensen. “Using RESTful web-services and cloud computing to create next generation mobile applications”. In: Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications. ACM. 2009, pp. 627–634.

[11] L. Li and W. Chou. “Design and Describe REST API without Violating REST: A Petri Net Based Approach”. In: 2011 IEEE International Conference on Web Services. 2011, pp. 508–515. doi: 10.1109/ICWS.2011. 54.

[12] L. Richardson and S. Ruby. RESTful Web Services. O’Reilly Media, 2008. isbn: 9780596554606. url: https://books.google.com.bd/books?id= XUaErakHsoAC.

[13] Amazon Web Services (AWS) - Cloud Computing Services. https : / / aws.amazon.com/. (Accessed on 09/03/2017).

[14] JSON Web Token (JWT). Date and time format - ISO 8601. https : //tools.ietf.org/html/rfc7519. (Accessed on 02/05/2017).

[15] Amazon Rekognition. https://aws.amazon.com/rekognition/. (Accessed on 05/07/2017).

Reference(Cont’d):

[16] The Python SQL Toolkit and Object Relational Mapper. https://www.sqlalchemy.org/. (Accessed on 05/07/2017).

[17] Amazon Elastic Load Balancing. https://aws.amazon.com/elasticloadbalancing/. (Accessed on 03/02/2017).

[18] Amazon Cloudwatch. https://aws.amazon.com/cloudwatch/. (Accessed on 04/04/2017).

[19] Color FERET Database. https://www.nist.gov/itl/iad/image-group/color-feret-database. (Accessed on 03/03/2017).

[20] S. Narula, A. Jain, and Prachi. “Cloud Computing Security: Amazon Web Service”. In: 2015 Fifth International Conference on Advanced Computing Communication Technologies. 2015, pp. 501–505. doi: 10.1109/ACCT. 2015.20.

[21] Haibin Ling et al. “Face verification across age progression using discriminative methods”. In: IEEE Transactions on Information Forensics and security 5.1 (2010), pp. 82–91.

[22] Unsang Park, Yiying Tong, and Anil K Jain. “Age-invariant face recognition”. In: IEEE transactions on pattern analysis and machine intelligence 32.5 (2010), pp. 947–954.

[23] KK Smitha, Tony Thomas, and K Chitharanjan. “Cloud based e-governance system: A survey”. In: Procedia Engineering 38 (2012), pp. 3816–3823.

[24] Erik Wilde and Cesare Pautasso. REST: from research to practice. Springer Science & Business Media, 2011.

[25] Anil K Jain and Stan Z Li. Handbook of face recognition. Springer, 2011.

Acknowledgments:

- Dr. Md. Whaiduzzaman, Associate Professor, IIT, JU.
- K M Akkas Ali, Director, IIT, JU.
- Jesmin Akhter, Associate Professor, IIT, JU.
- Dr. Mohammad Abu Yousuf, Associate Professor, IIT, JU.
- Dr. Mohammad Shahidul Islam, Assistant Professor, IIT, JU.
- Ministry of Science and Technology.