

## APPENDIX O

Please also explore BCC's Preliminary Roadmap and the referenced materials therein. The Abstract for this RFP is also included by reference herein. This work could extend to a number of ICAM categories of software and network customization including the sprints for OIDC IdP Provider and ICAM Solution functions, features, and extensions described below. Interested Parties ought to look at BCC's RFP Abstract, Executive Summary and Preliminary Roadmap next. **BCC's Contact** and Official Website: [biometriccare.com](http://biometriccare.com)  
 Contract Location: California. (8:30 a.m. - 5:30 p.m. PST); email: [Steve@biometriccare.com](mailto:Steve@biometriccare.com) - Please add "RFP" to the Subject line.

No	Primary Function	Example Software: Name/Link	Licensing	Criticality	Pre-Tested /Trained	Code	Category-Model
1	Development Project Mgmt	<a href="#">andygrunwald/go-jira</a> <a href="#">cptactionhank/docker-atlassian-jira-software</a>	FOSS MIT	Critical		Go Ruby	Engineering Patterns and Versioning
2	Biometric ID w Biometric Encryption or Cancellable Biometrics	<a href="#">Mariuki/AutomatedFuzzyVaultFingerprint</a> <a href="#">jwoogerd/fuzzy_vault</a>	FOSS MIT  Unclear	High Priority	compare articles...	MATLAB  Python	Finger-print  Finger-piint
3	Face Recognition	<a href="#">timesler/facenet-pytorch</a>	FOSS MIT	Critical	pretrained	Python	Face Recognition

		<a href="#">davidsandberg/facenet</a>  <a href="#">serengil/deepfacei</a>  <a href="#">HuangYG123/CurricularFace</a>  <a href="#">justadudewhohacks/face-api.js</a>	FOSS MIT  AWS via Dockers?  FOSS MIT  FOSS MIT  FOSS MIT		over 99%  pretrained over 99%	Python & MATLAB  Python & Jupyter Notebook  Python  TypeScript	Map Face to patient 's First Name, Last Name and DOB  JavaScript API for face detection and face recognition in the browser and nodejs with tensorflow.js
3 bis	Extends IOS FaceID and TouchID for subjects with iPhones	<a href="#">rushisangani/BiometricAuthentication</a>	FOSS MIT	Desired		Swift, Ruby, Objective-C	Use Apple FaceID or TouchID authentication in your app using BiometricAuthentication.
4	Authentication: OIDC IdP	<a href="#">node oidc-provider</a>  <a href="#">node oidc-provider ^6.5.0</a>	FOSS MIT  FOSS MIT	Critical	3rd-Party init OP	Java Script Environments	OIDC for node.js servers

					Not above		
	Biometric System Performance Evaluation	<a href="#">manuelaguadomtz/pyeer</a>	FOSS MIT	Critical		Python	
5	Interoperability	FHIR R4.0	Open Standard	Critical		TLS/mTLS	Transport compatible with EMRs
6		<a href="https://github.com/CareEvolution/Public/tree/master/PDemo">https://github.com/CareEvolution/Public/tree/master/PDemo</a>		Critical	Demographic Locator-compatible with EPIC and Cerner		See also NHS
7	Federation	UDAP and Carequality	FAST HL7 & ONC	High Priority			
8	Mobile Application Development Platform	<a href="#">Android™ App Development</a> <a href="#">iOS™ App Development</a> Cross-platform Development: Open Source Platform (under MIT License) <a href="#">Microsoft Xamarin Platform</a> <a href="#">Kony App Platform™</a>	See Links Kony is now Temenos	Critical			GAIN & NHS

		Kony Mobility Platform					
		Kony Cloud, etc					

---

i Face recognition models. Deepface is a hybrid face recognition package. It currently wraps many state-of-the-art face recognition models: [VGG-Face](#) , [Google FaceNet](#), [OpenFace](#), [Facebook DeepFace](#), [DeepID](#), [ArcFace](#) and [Dlib](#). The default configuration uses VGG-Face model.

```
models = ["VGG-Face", "Facenet", "Facenet512", "OpenFace", "DeepFace", "DeepID", "ArcFace", "Dlib"]
```

```
result = DeepFace.verify(img1_path = "img1.jpg", img2_path = "img2.jpg", model_name = models[1])
```

```
df = DeepFace.find(img_path = "img1.jpg", db_path = "C:/workspace/my_db", model_name = models[1])
```

FaceNet, VGG-Face, ArcFace and Dlib *overperforms* than OpenFace, DeepFace and DeepID based on experiments. Supportively, FaceNet /w 512d got 99.65%; FaceNet /w 128d got 99.2%; ArcFace got 99.41%; Dlib got 99.38%; VGG-Face got 98.78%; DeepID got 97.05; OpenFace got 93.80% accuracy scores on *LFW data set* whereas human beings could have just 97.53%.