

Conversational Biometrics Group  
IBM Research



# ***Conversational Biometrics:*** ***Breaking the sound barrier of*** ***secure voice applications***

**Ganesh N. Ramaswamy**  
**Manager, Conversational Biometrics**  
**IBM T. J. Watson Research Center**  
**Yorktown Heights, New York**  
**ganeshr@us.ibm.com**  
**<http://www.research.ibm.com/CBG>**



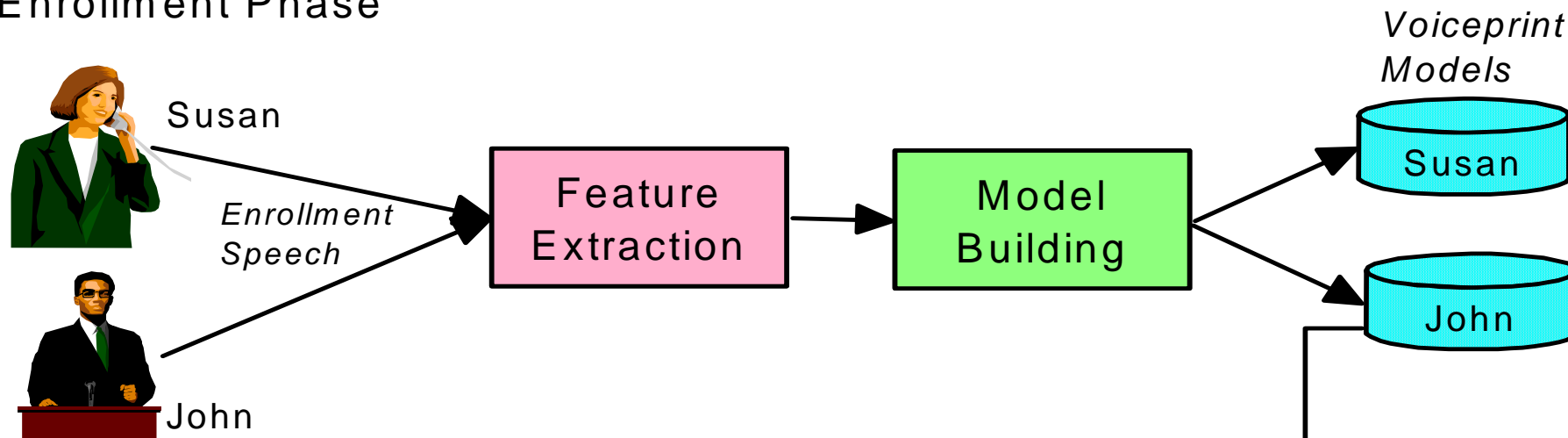
BCC 2003

## Conversational Biometrics (CB)

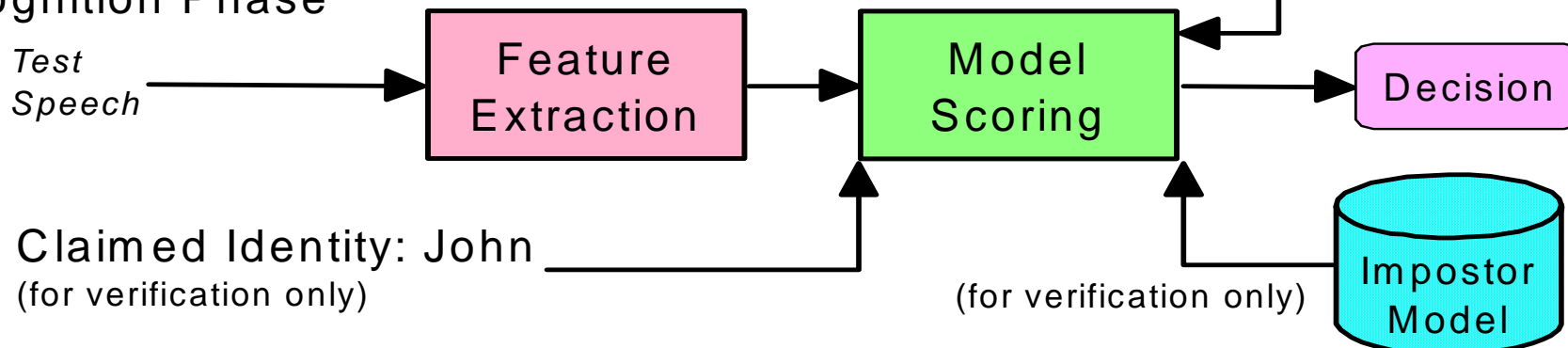
- Combine acoustic (text-independent) voiceprint match with conversational knowledge match for voice-based speaker recognition with high accuracy & flexibility
- Departure from static combination of voiceprint and knowledge
- Single conversational interface consisting of:
  - Acoustic text-independent speaker verification
  - Conversational knowledge verification (speech recognition, natural language understanding, dialog management, text-to-speech synthesis)
  - Programmable policy management
- Distributed architecture with support for “plug-in” engines to include additional authentication modalities

# Acoustic Text-Independent Speaker Recognition

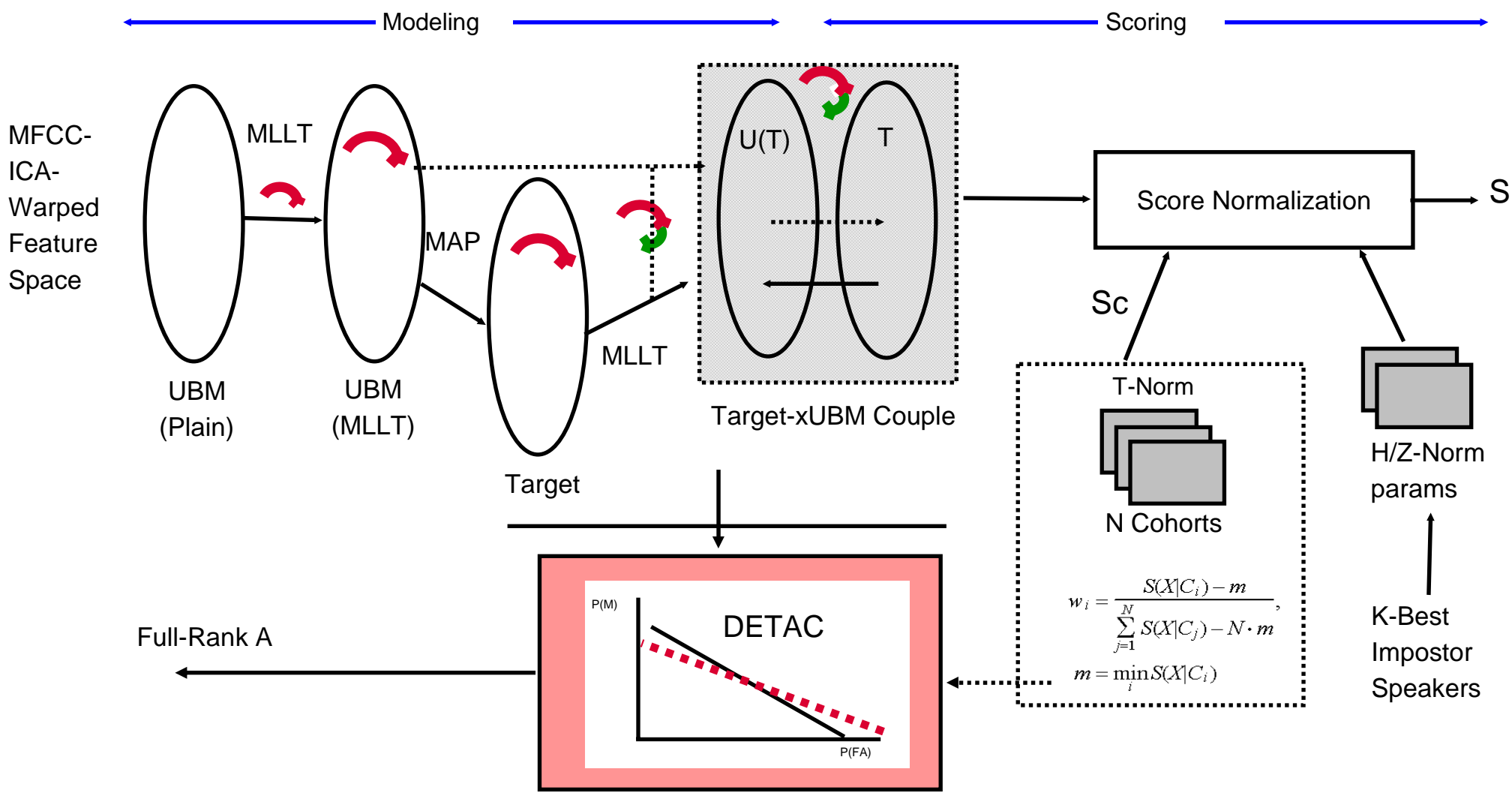
## Enrollment Phase



## Recognition Phase



# Acoustic Text-Independent Speaker Recognition

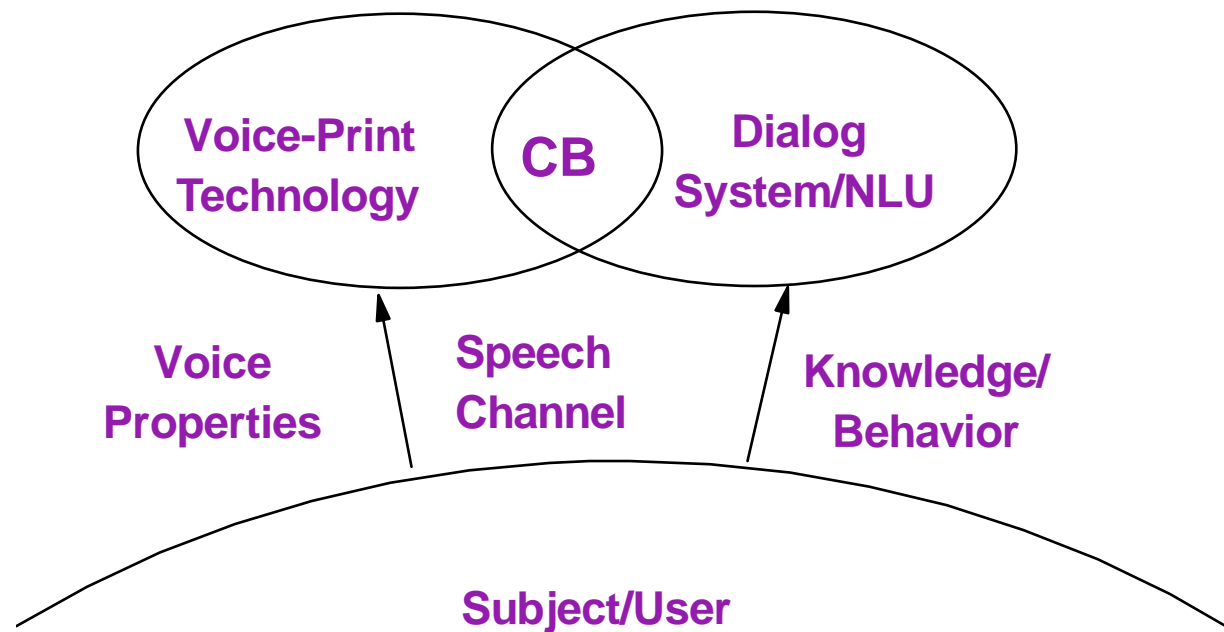


# Acoustic Text-Independent Speaker Recognition

- Voiceprint enrollment
  - Feature set: Mel-Frequency Cepstral Coefficients (MFCC), ICA, non-linear warping
- Modeling
  - Clustering: LBG/KMeans or Eigenvector-based
  - Grainmodel: Gaussian Mixture Model (GMM), MLLT feature space
  - Multigrain structure: multiple coarseness levels
  - Adaptation from a speaker-independent model
  - Internal HMM phonetic (open-loop) labeling, grammar-based decoding/alignment
- Recognition
  - Likelihood scoring: Pickmax on the GMM-tree
  - Imposter models: UBM, Cohort from BG pool, cohort-discriminative linear transforms
  - Scoring: log-likelihood ratio test, on-the-fly BG score weighting, BG pruning
  - Discriminative model/feature transform (DETAC)
  - Channel norms: adaptive T,Z, and H
  - Confidence scores

## Conversational Biometrics (CB)

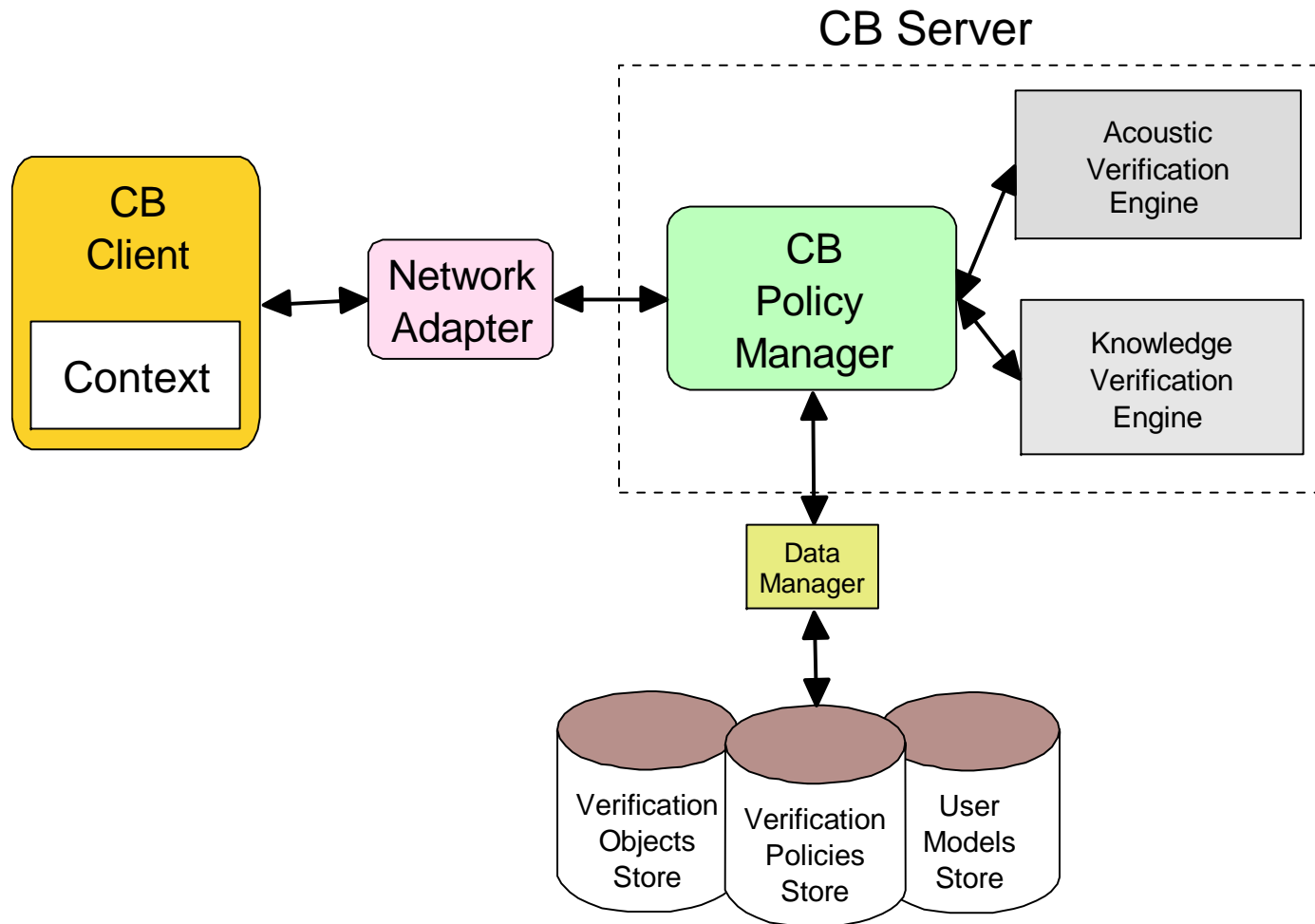
- Combine acoustic (text-independent) voiceprint match with randomized conversational knowledge verification
  - Single conversational interface for both voiceprint and knowledge verification
  - Verification process managed by a programmable policy manager



# Conversational Biometrics (CB) Server Features

- Acoustic Speaker Verification
  - Text-independent - no constraints on voice input.
  - Verification & identification on same engine. Scalable to support large population
  - Integrable with conversational applications
  - Multiple operating systems (AIX, Linux, Windows)
  - Multi-device feature (telephony, desktop, etc.)
- Integrated Conversational Biometrics
  - Programmable policy management with dynamic verification policies
  - Flexible operation modes: acoustic only, knowledge only & combination
    - Optionally accepts external scores (from alternate authentication modules)
  - Customizable hierarchical topic structure, support for dynamic answers
  - Application independent client-server architecture
  - Stateless ("one-shot") servers
    - Private non-exposed context stored on client side
  - Flexible data access

# CB Client-Server Architecture



## CB Policy Manager

- Departure from static combination of voiceprint and knowledge
- Extremely flexible way to tradeoff convenience and security.
- Broad interpretation of Conversational Biometrics. Accommodates application-specific, transaction-specific & user-specific requirements
- Dynamic combination of multiple authentication modalities: voice, knowledge, caller-id, etc.
  - Manage trade-off between security and convenience, achieving arbitrarily low error rates with combination of multiple verification objects
  - Customizable hierarchy of verification objects, including dynamic objects
  - Support for plug-in engines & external scores from alternate verification engines
  - Incremental authentication, multiple security levels
- Components of the policy manager:
  - Verification objects (and associated verification engines)
  - Verification policies
  - User models

## CB Policy Manager

- CB policies:
  - FSM implementation using XML.
  - One or more verification challenges at each state
  - Dynamically (random, semi-random, fixed) determined based on current context, transaction requirements and user preferences. Exact or approximate match.
  - Decisions (accept, reject, continue) based on current context
  - Offers complete programming and customization of the verification process
- CB policies operate on the session *context*.
  - Session context contains: user name, current state within policy, history of verification objects invoked, scores related to invocation, transaction-specific, user-specific and other logical/physical variables
  - Context may include scores from external verification sources
  - Context may also include additional user defined variables
  - Context updated on every turn, rules applied to context resulting in one of three possibilities: accept, reject, or continue

## Example Registry of Verification Objects

```

<verification_object_base>
  <object name="DOB"
    Engine="Knowledge"
    Type="QA"
    Prompt="What is your Date of Birth?"
    Perplexity="10" ></object>
  <object name="CALLER_ID"
    Engine="Telephony"
    Type="Caller_ID"
    Prompt=none
    Perplexity="20" ></object>
  <object name="VOICE_PRINT"
    Engine="Voiceprint"
    Prompt=none
    Perplexity="1000" ></object>
  <object name="COLOR"
    Engine="Knowledge"
    Type="QA"
    Prompt="What is your favorite color?"
    Perplexity="5" ></object>
  <object name="CAR_COLOR"
    Inherit_from="COLOR"
    Prompt="What is the color of your car ?"></object>
  <object name="CUR_BALANCE"
    Engine="Knowledge"
    Type="APP_NUM"
    Prompt="What is current the approximate balance
in your account?"
    Perplexity="100"></object>
  <object name="LAST_TRANSACTION_DATE"
    Engine="Knowledge"
    Type="APP_STR"
    Prompt="What is the date of your last transaction??"
    Perplexity="100"></object>
</verification_object_base>

```

## Example User Model

```
<user_model name="JOHN_DOE">
  <objects>
    <object name="CALLER_ID"
      Answer="914-945-3000"
      Preference="20"></object>
    <object name="DOB"
      Answer="08-02-1975"
      Preference="20"></object>
    <object name="COLOR"
      Answer="blue"
      Preference="10"></object>
    <object name="CAR_COLOR"
      Answer="red"
      Answer="beige"
      Preference="30"></object>
    <object name="VOICE_PRINT"
      Filename="john_doe.vpr"
      Preference="30"></object>
    <object name="CUR_BALANCE"
      Preference="10"></object>
    <object name="LAST_TRANSACTION_DATE"
      Preference="10"></object>
  </objects>
</user_model>
```

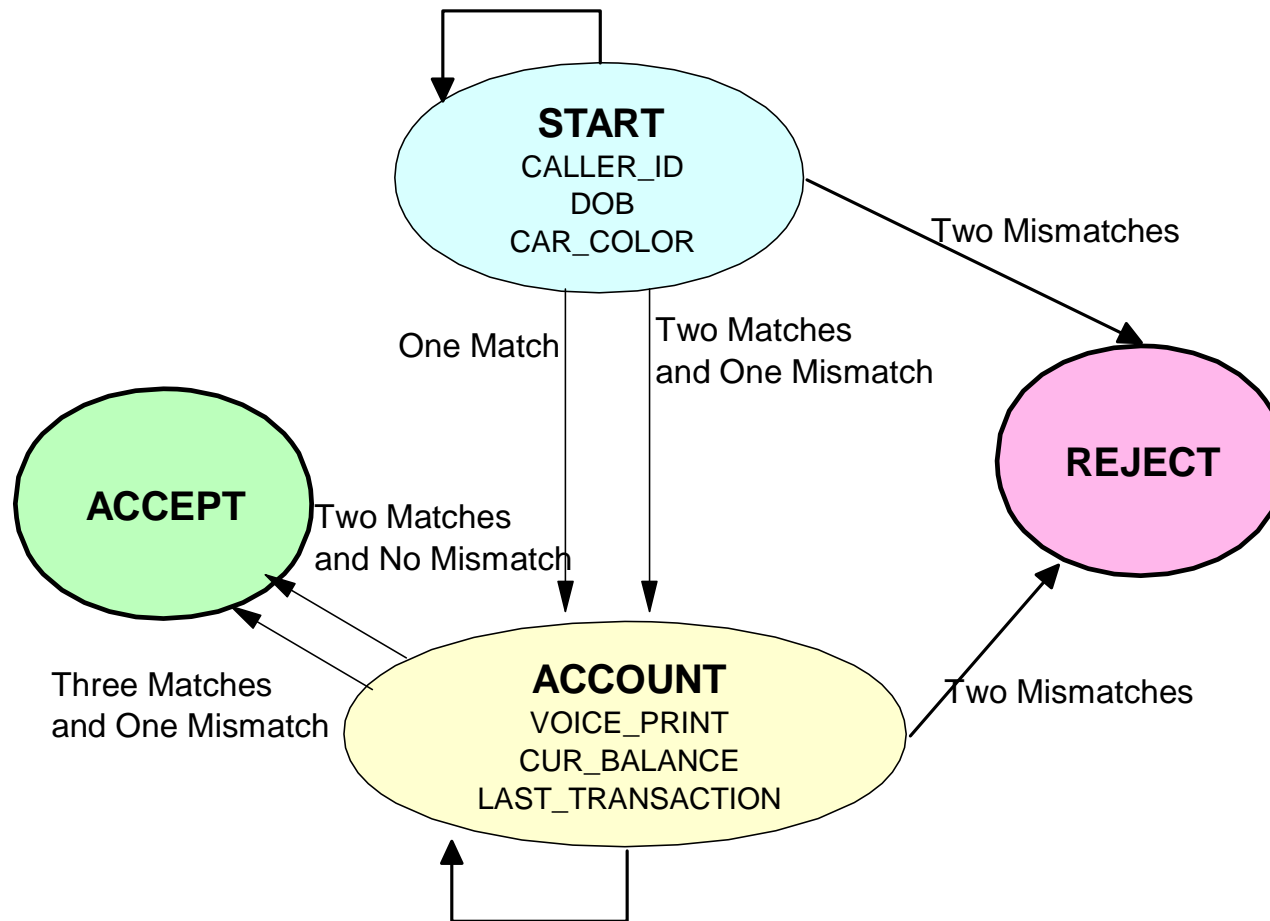
# Example Policy Specification in XML

```

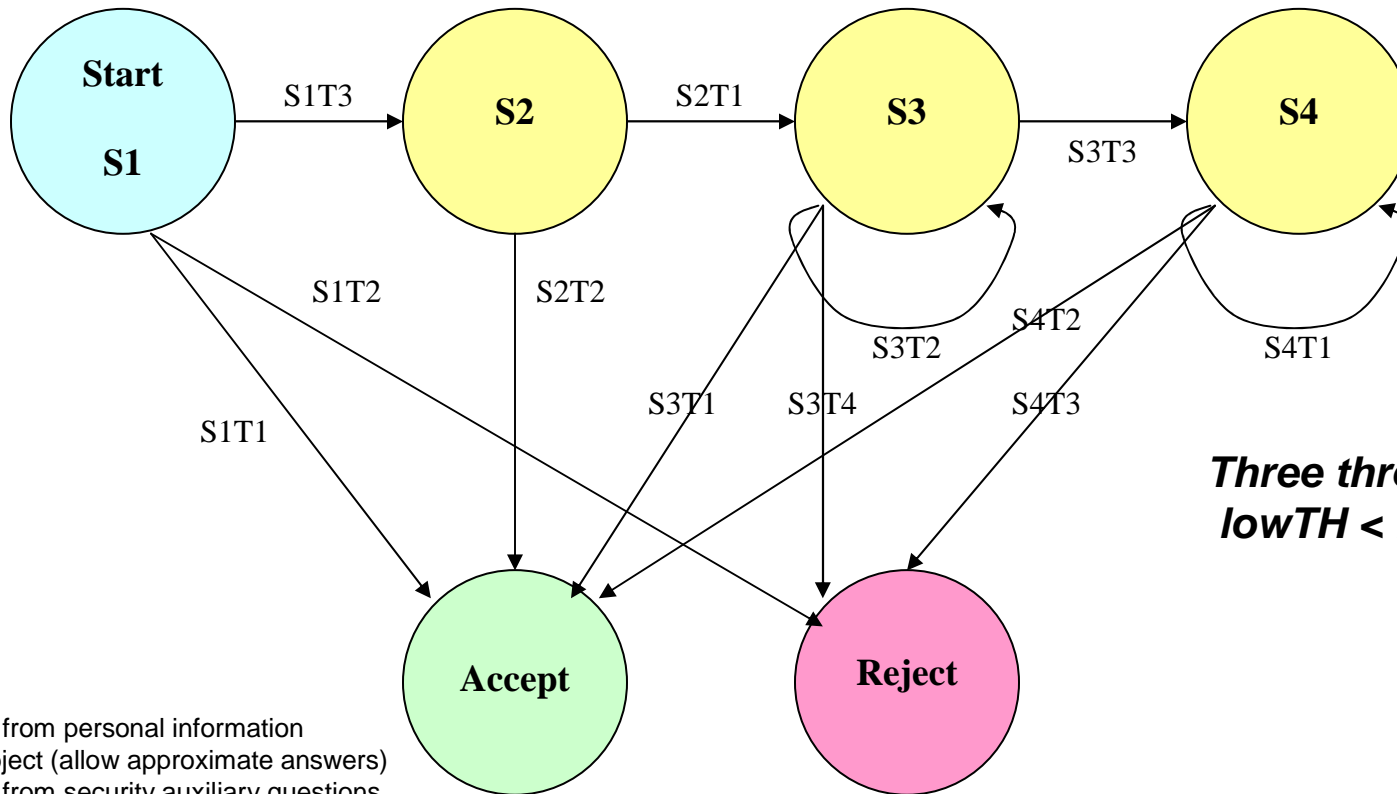
1. <policy name="SIMPLE_BANK_POLICY">
2.   <variables>
3.     <default var="curBalance"          val="1"          type="float"></default>
4.     <default var="minVoiceprintScore" val="0"          type="float"></default>
5.     <default var="lastTransactionDate" val="01-01-2100" type="string"></default>
6.   </variables>
7.   <conditions>
8.     <condition name="ONE_OK"          exp="_curObjectNum=1 & _curWrongNum=0"></condition>
9.     <condition name="TWO_OK_ONE_BAD"  exp="_curObjectNum=3 & _curWrongNum=1"></condition>
10.    <condition name="TWO_BAD"         exp="_curWrongNum = 2"></condition>
11.    <condition name="TWO_OK_NO_BAD"   exp="_curObjectNum=2 & _curWrongNum=0"></condition>
12.    <condition name="THREE_OK_ONE_BAD" exp="_curObjectNum=4 & _curWrongNum=1"></condition>
13.    <condition name="CUR_BALANCE_TEST" exp="_answerFloat > curBalance*0.95 &
    _answerFloat < curBalance*1.05" ></condition>
14.    <condition name="VOICE_PRINT_TEST" exp="_answerFloat > minVoiceprintScore"></condition>
15.    <condition name="DATE_TEST"      exp="_answerString = lastTransactionDate" ></condition>
16.  </conditions>
17.  <states>
18.    <state name="_ACCEPT_">
19.    </state>
20.    <state name="_REJECT_">
21.    </state>
22.    <state name="_START_">
23.      <objects>
24.        <object name="CALLER_ID"      weight="100"></object>
25.        <object name="DOB"            weight="10"></object>
26.        <object name="CAR_COLOR"     weight="10"></object>
27.      </objects>
28.      <switch>
29.        <case condition="ONE_OK"      target="ACCOUNT" ></case>
30.        <case condition="TWO_OK_ONE_BAD" target="ACCOUNT" ></case>
31.        <case condition="TWO_BAD"    target="_REJECT_"></case>
32.      </switch>
33.    </state>
34.    <state name="ACCOUNT">
35.      <objects>
36.        <object name="VOICE_PRINT"    condition="VOICE_PRINT_TEST"></object>
37.        <object name="CUR_BALANCE"    condition="CUR_BALANCE_TEST" ></object>
38.        <object name="LAST_TRANSACTION_DATE" condition="DATE_TEST" ></object>
39.      </objects>
40.      <switch>
41.        <case condition="TWO_OK_NO_BAD" target="_ACCEPT_"></case>
42.        <case condition="THREE_OK_ONE_BAD" target="_ACCEPT_"></case>
43.        <case condition="TWO_BAD"      target="_REJECT_"></case>
44.      </switch>
45.    </state>
46.  </states>
47. </policy>

```

# Example CB Policy (1)



## Example CB Policy (2)



**Three thresholds:**  
 $lowTH < medTH < highTH$

S1: Randomize from personal information  
 S2: Dynamic object (allow approximate answers)  
 S3: Randomize from security auxiliary questions  
 S4: Randomize from personal information

S1T1: Correct answer, very high score  
 S1T2: Wrong answer, very low score  
 S1T3: Correct answer, score below very high  
 OR Wrong answer, score above very low  
 S2T1: Less than 2 correct answers, high score OR low score  
 S2T2: 2 or more correct answers, high score

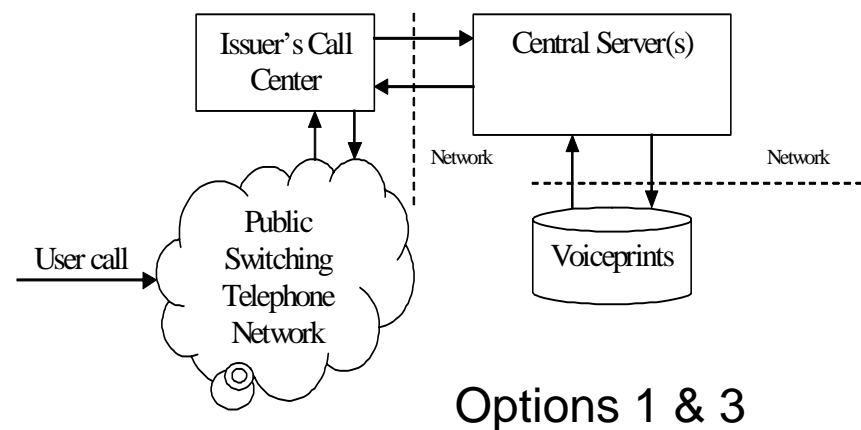
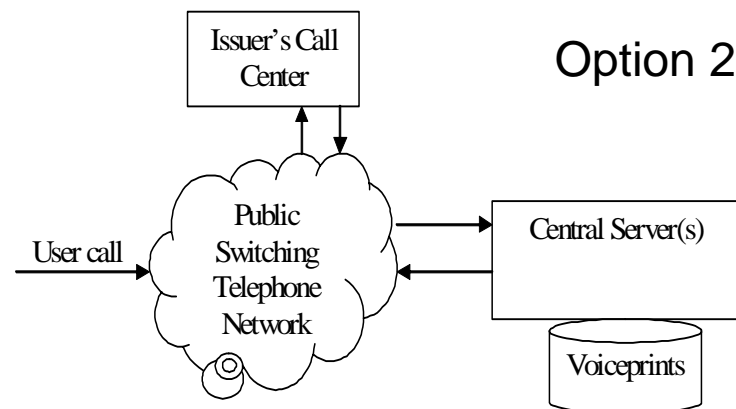
S3T1: Less than 3 wrong answers, high score  
 S3T2: Less than 3 wrong answers, low score  
 S3T3: 3 or more wrong answers, high score  
 S3T4: 3 or more wrong answers, low score  
 S4T1: less than 2 correct answers, high score  
 OR more than 2 correct answers and score between lowTH & medTH  
 S4T2: 2 or more correct answers, high score  
 S4T3: less than 2 correct answers and low score OR very low score

## Example Application: Credit-Card Fraud Prevention

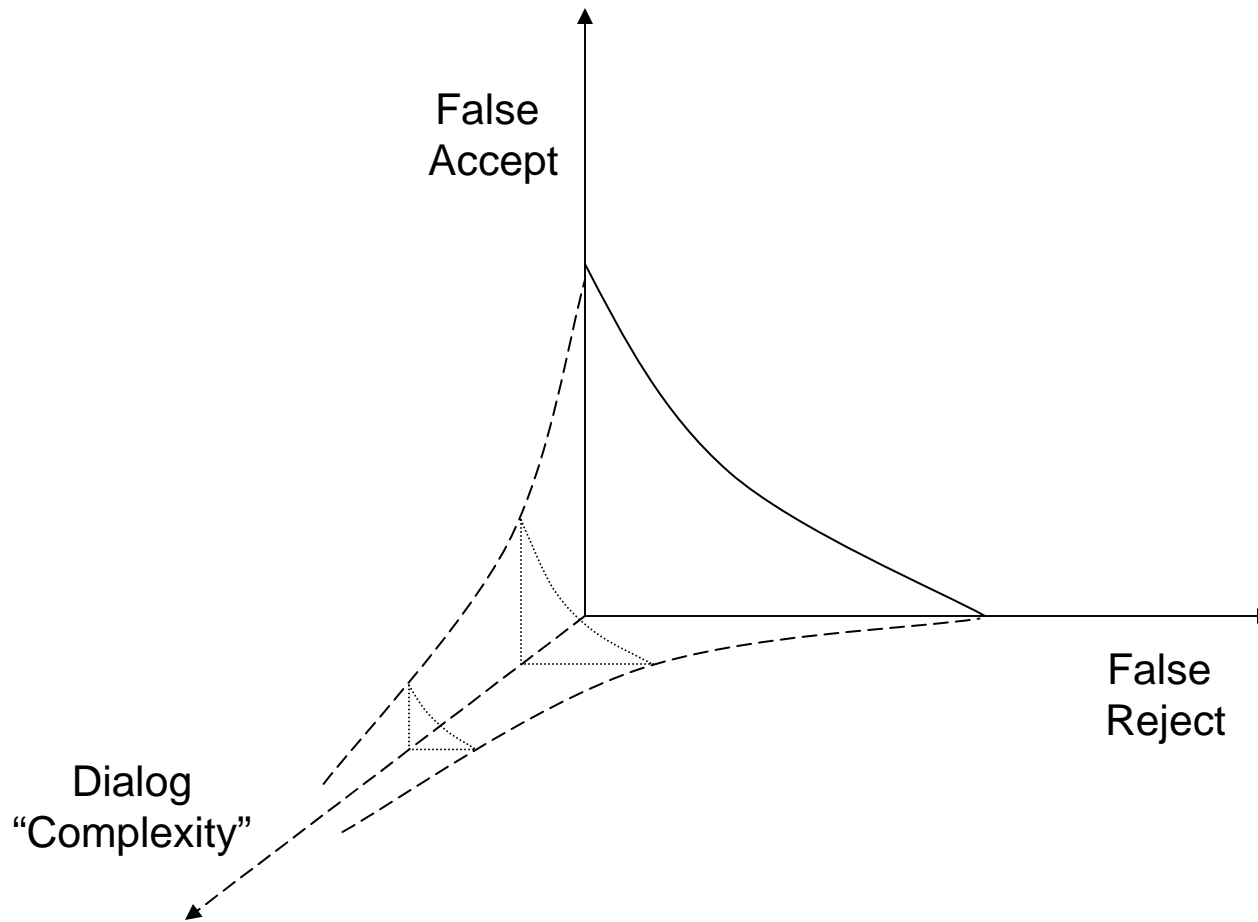
- Conversational Biometrics solution for credit-card fraud & identity theft prevention, fraud analysis, detection of counterfeit cards, detection of frequent fraudsters and mitigation of operational risk
- Requires minimal / no change to existing practices
- Secure and reliable enrollment protocol
  - Voiceprint enrollment: performed at activation of new card issuance
  - Knowledge enrollment: use existing account information
- Verification protocol
  - Automatic authentication of voice-based transactions
  - Selective explicit voice-based authentication for high-risk or unusual (non-voice) transactions
  - Outbound calls for proactive fraud & identity theft prevention
- Online and offline fraud prevention & analysis
  - Integration with existing fraud detection and analysis systems
  - Adds additional input to investigator cases
  - Creation of frequent fraudster databases

## Solution Options

- Option 2: Local Install at the bank location
- Options 1 & 3: Distributed client-server implementation
  - Switching through the telephone network (option 1) or through IP network (option 3)
  - Possible hosting of “third-party authentication” services



# Policy Management and Error Tradeoff



## Why Conversational Biometrics ?– Users will not need to change existing habits making adoption faster.

- Accuracy

  - Arbitrarily low error with combination of voiceprint & randomized knowledge match

  - Incremental authentication: collect more data for higher confidence

  - Speaker tracking, detect (unauthorized) speaker changes

- Flexibility

  - Customizable verification policies

  - Customizable hierarchy of verification objects, including dynamic objects

  - Continuous authentication, background authentication

- Convenience

  - Non-intrusive text-independent natural language input, same voice analyzed twice

  - Continuous enrollment: on-line adaptation of acoustic speaker models

  - Qualified identity management to streamline customer service

# The Big Picture

## Quote from a fortune cookie:

*To know the mind of a man, listen to his words*

