



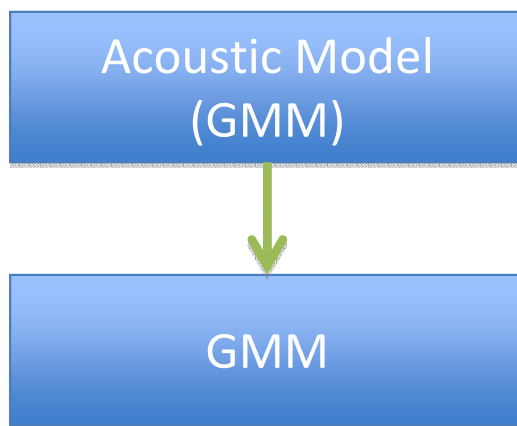
Acoustic models in Kaldi

Acoustic models in Kaldi

- Support for standard ML-trained models
 - Linear transforms like LDA, HLDA, MLLT/STC
 - Speaker adaptation with fMLLR, MLLR
 - Support for tied-mixture systems initially discussed
- Support for SGMMs
 - Speaker adaptation with fMLLR (single transform) in addition to speaker subspaces
- Modular code, can be easily extended



Overview of AM Classes



```
std::vector< GMM* >
```



“knows about”



GMM

- Gaussians represented using *natural parameters*.
 - For efficient likelihood evaluation

GMM

- Gaussians represented using *natural parameters*.
 - For efficient likelihood evaluation

$$\begin{aligned}\log f_{\text{Gauss}}(\mathbf{x}) &= K_1(\boldsymbol{\Sigma}) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \\ &= K_1(\boldsymbol{\Sigma}) - \frac{1}{2} [\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}] \\ &= K_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}\end{aligned}$$

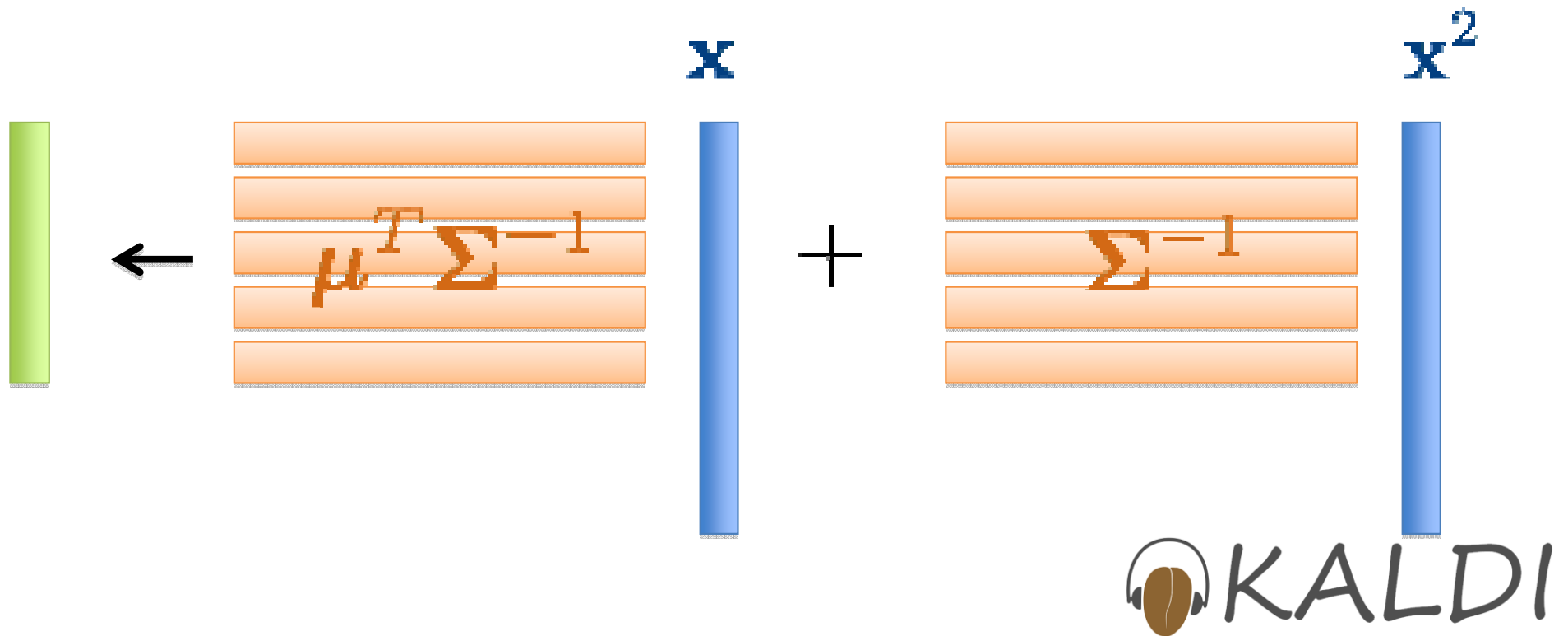
GMM

- Gaussians represented using *natural parameters*.
 - For efficient likelihood evaluation

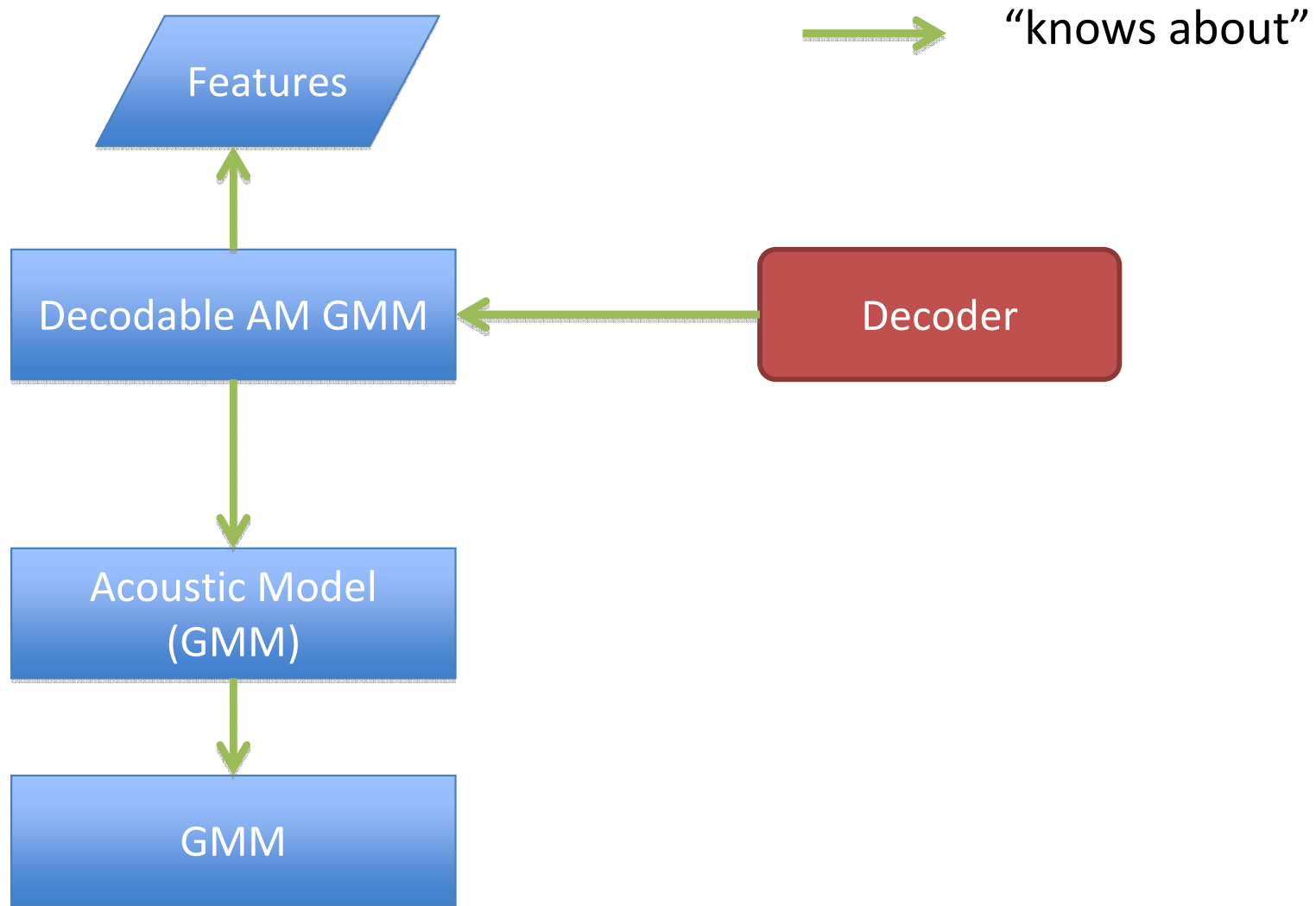
$$\begin{aligned}\log f_{\text{Gauss}}(\mathbf{x}) &= K_1(\boldsymbol{\Sigma}) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \\ &= K_1(\boldsymbol{\Sigma}) - \frac{1}{2}[\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}] \\ &= K_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \boxed{\quad} \mathbf{x} - \mathbf{x}^T \boxed{\quad} \mathbf{x}\end{aligned}$$

GMM

- Likelihood calculation done in 2 matrix-vector multiplications.
 - Optimized BLAS routines can be used.



Overview of AM Classes

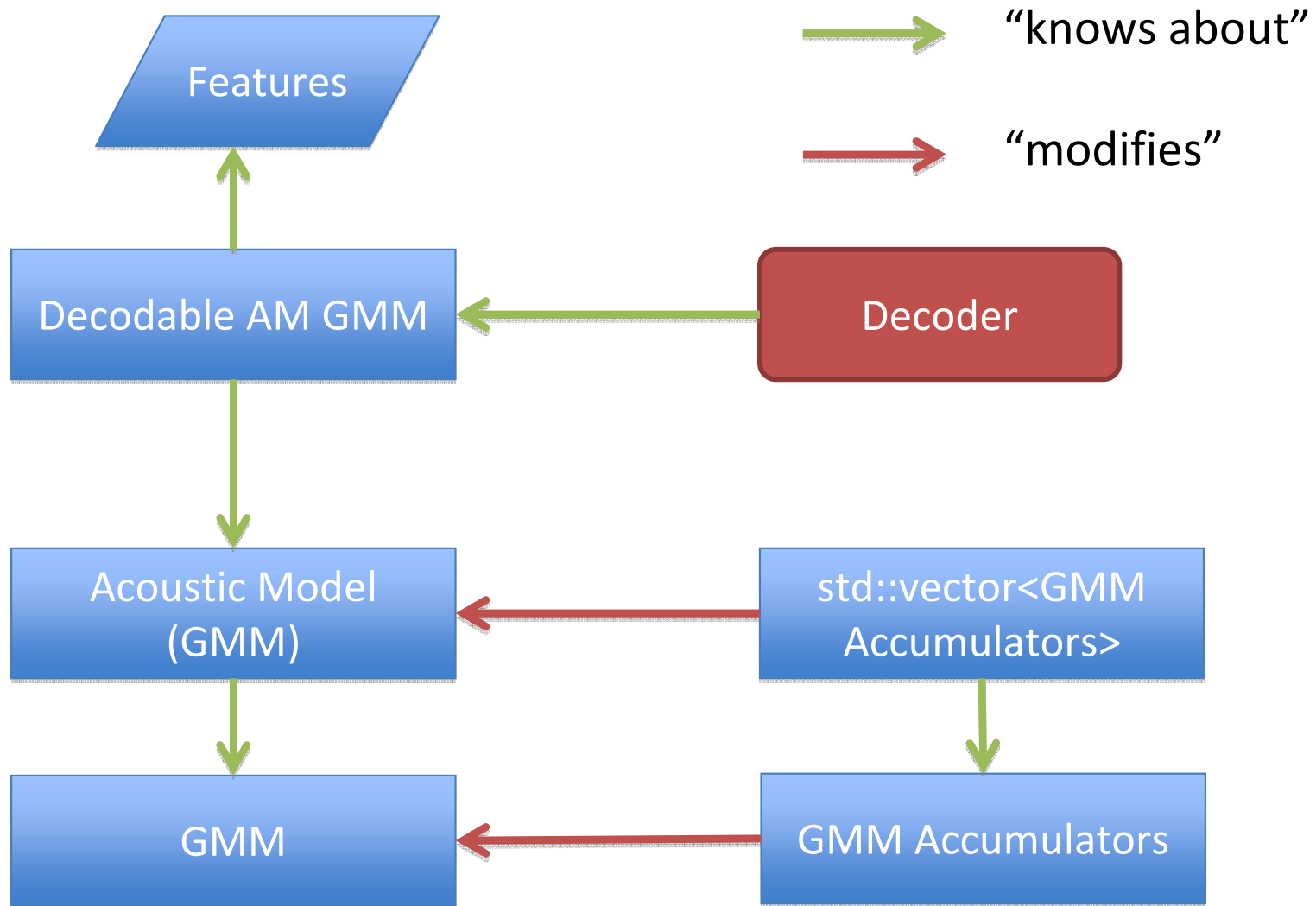


The Decodable Interface

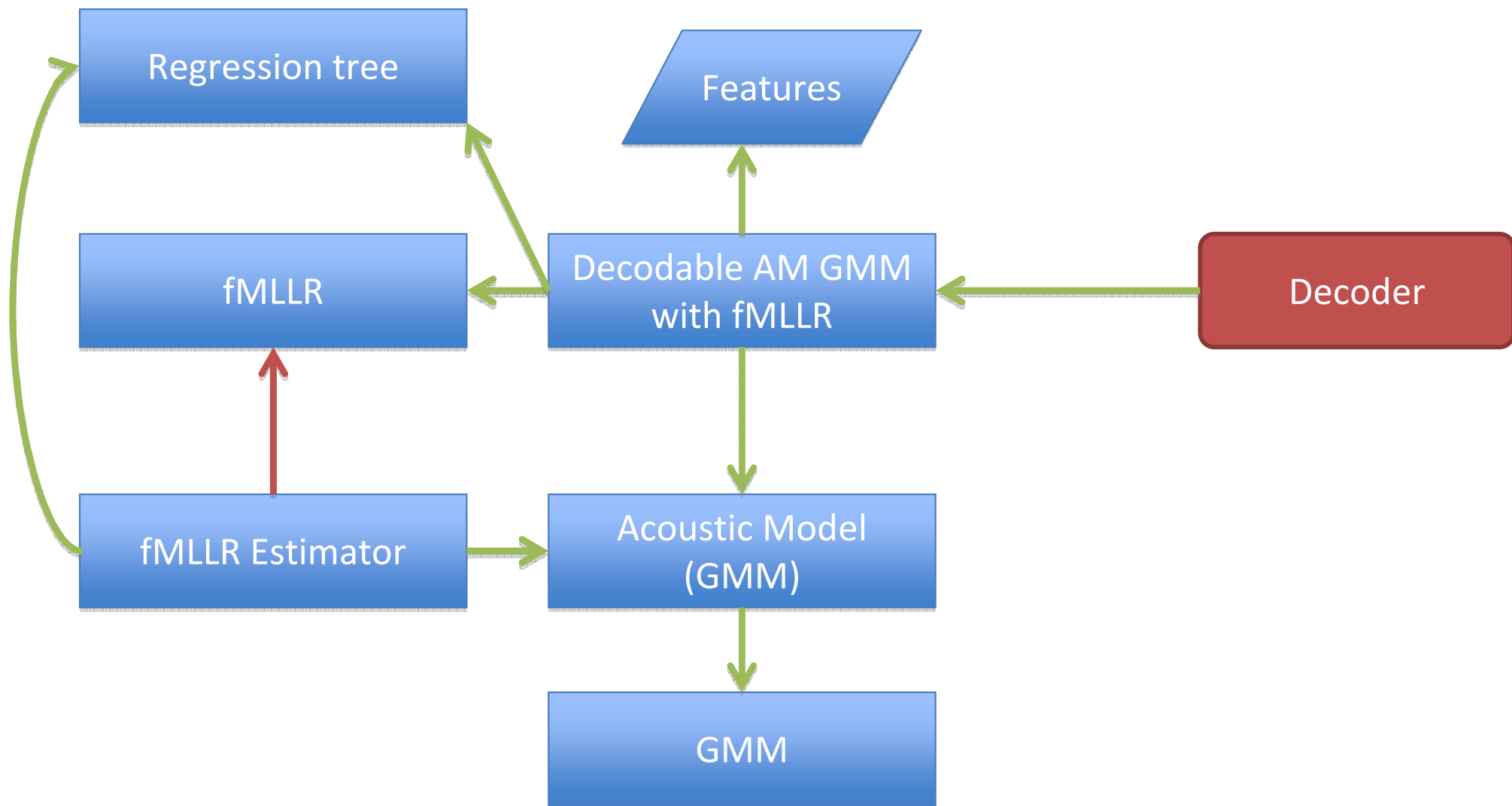
```
class DecodableInterface {  
public:  
    // Returns the log likelihood (negated in the decoder).  
    virtual BaseFloat LogLikelihood(int32 frame, int32 index) = 0;  
  
    // Frames are one-based.  
    virtual bool IsLastFrame(int32 frame) = 0;  
  
    /// Indices are one-based (compatibility with OpenFst).  
    virtual int32 NumIndices() = 0;  
  
    virtual ~DecodableInterface() {}  
};
```



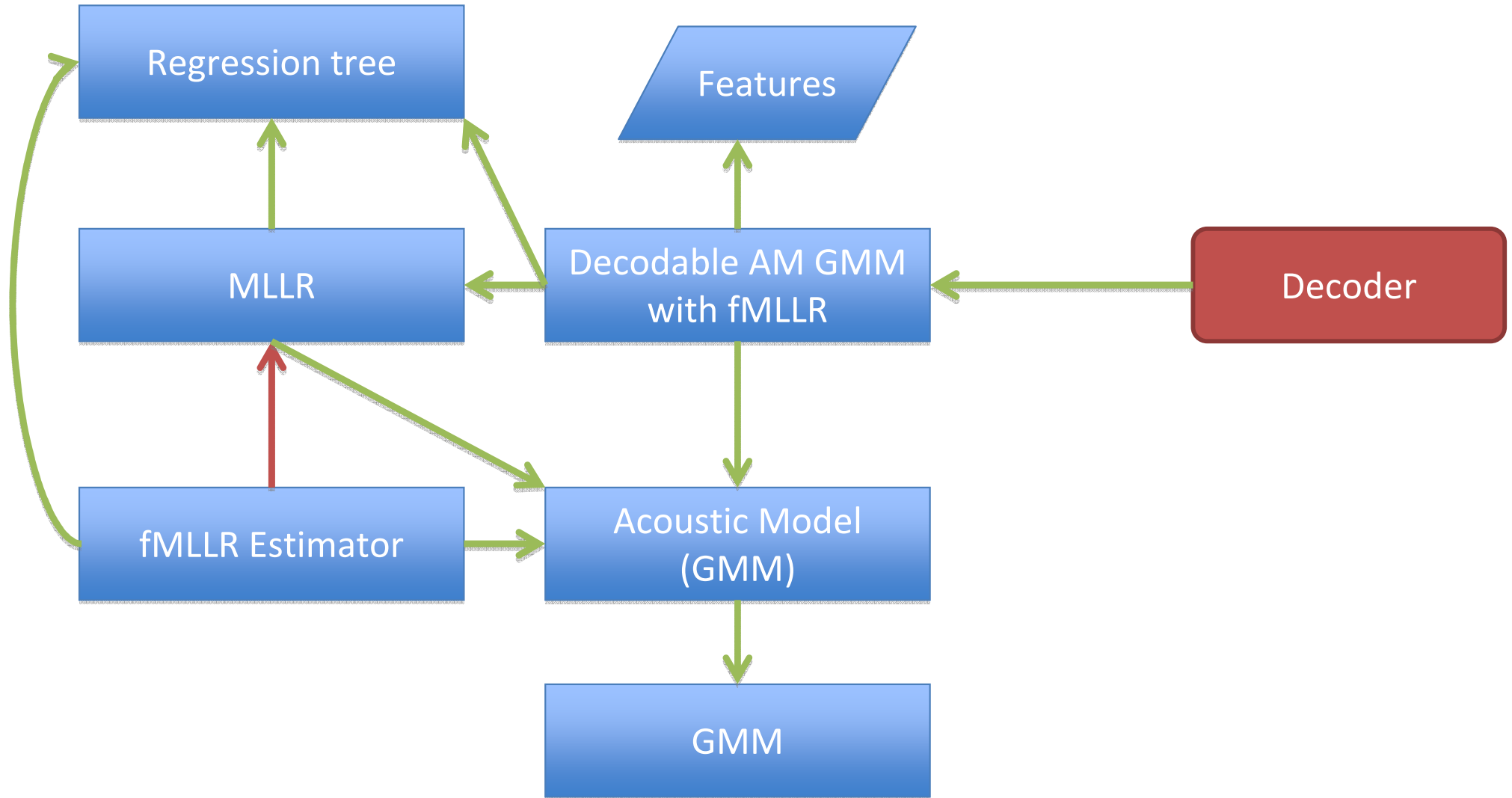
Overview of AM Classes



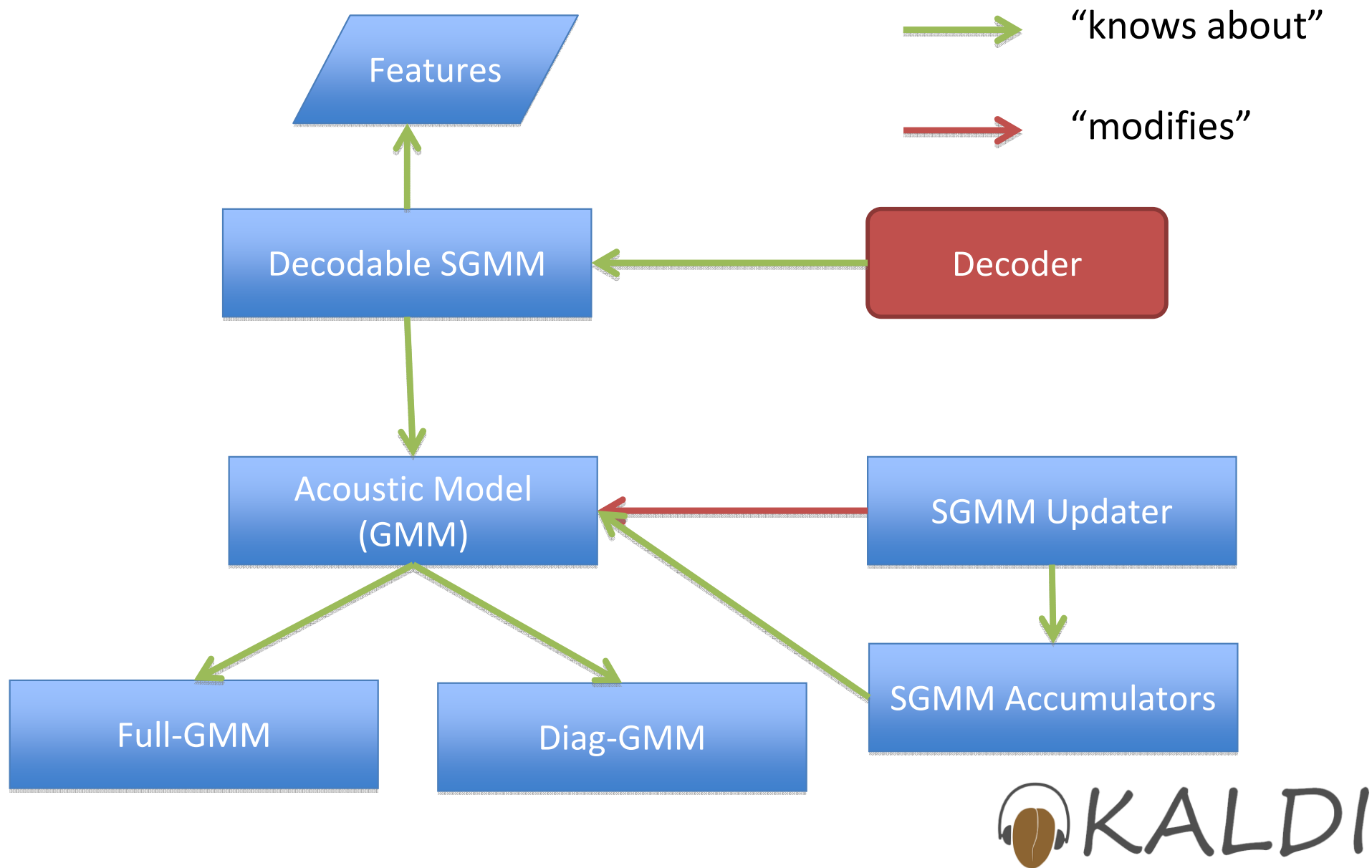
AM Classes with fMLLR



AM Classes with MLLR



Overview of SGMM Classes



Things to do next

- fMLLR basis for SGMMs
- The “Symmetric” extension to SGMMs
- Discriminative training code planned for this summer
 - Need lattice generation
- Thoughts on multiple feature transforms