Structured Content-Independent Scalable Meta-formats (SCISM) for Media-Type Agnostic Transcoding

Debargha Mukherjee, Amir Said

Presented by: Giordano Beretta

Compression and Multimedia Technologies Project
Imaging Technology Department, ISL
Hewlett Packard Laboratories

Outline

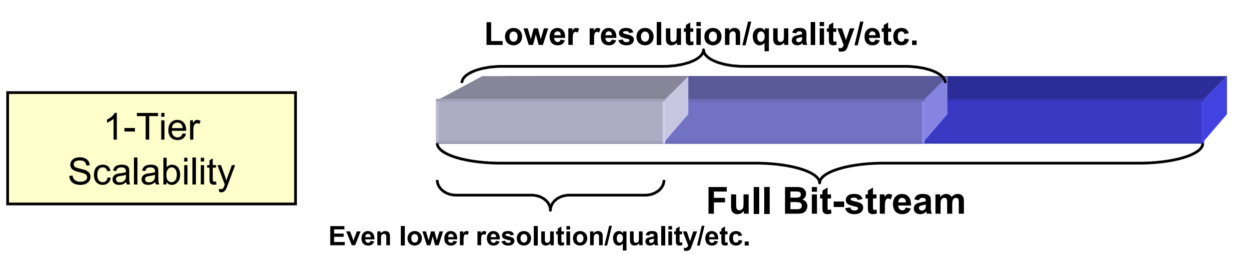
- Introduction
- Scalable Bit-streams Review
- Universal Meta-Format: SCISM
- Capabilities & Preferences Specifications
- Transcoding Operation
- Example Scenarios
- Conclusion

Introduction: Internet is Heterogeneous

- Diversity over the Internet is here to stay
  - Variety of Bandwidths
    - Wired: 56K/DSL/T1/T3/OC-X; Wireless: 2.5G/3G/4G.
  - Variety of Display Resolutions, Graphics Hardware
    - PDA/Notebook/Desktops/Visualization-Workstations
  - Variety of Processing Powers.
- The vision – 'Seamless ease of Media Use'
  - Won't materialize without adequate focus on media content adaptation.

Introduction: Media Adaptation Approaches

- Maintain multiple-versions of media:
  - Drawback: Storage & Bandwidth wastage
- Scalable bit-streams promise a better solution:
  - Smaller subsets of the whole bit-stream produce representations at lower quality, resolution, etc.
    - Can potentially eliminate all redundancies.
  - Simple transcoding from higher to lower version.



Introduction: Types of Scalability

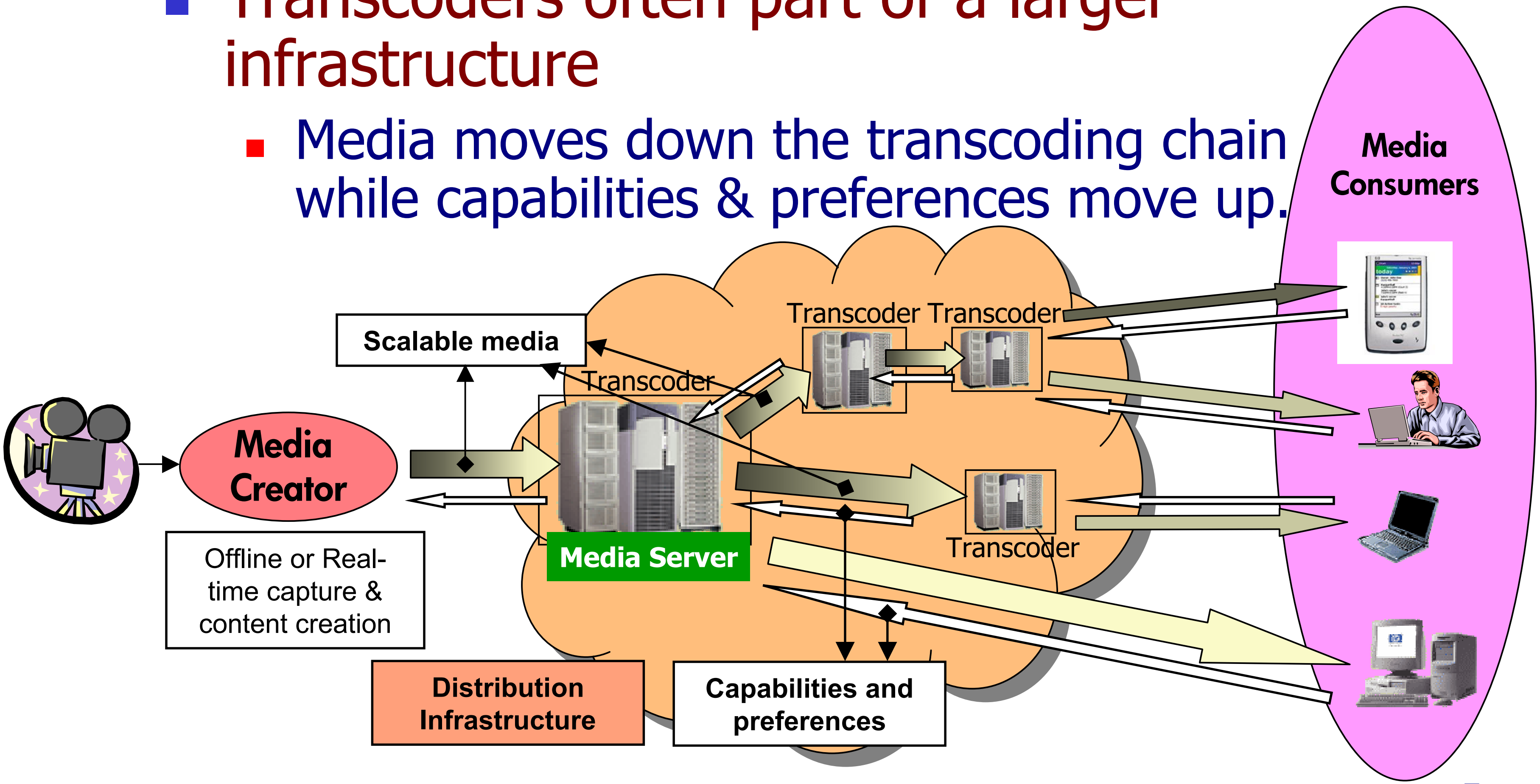
- Known types of scalability:
  - SNR – Applies to most types of media
  - Resolution – Image, Video, 3D media
  - Temporal – Video, Image sequences
  - Interactivity – New interactive media
- New types of scalability will evolve with new media – 3D, 4D, Aromas etc.
- Scalability types may be combined

Introduction: Scalable Media Communication

- Transcoders needed to reap the benefits of a scalable bit-stream.
  - Transcoders generate several versions from the same parent scalable bit-stream.
- Transcoders can be included in:
  - *Media servers* – media originates from here
  - *Mid-stream transcoding/routing servers,* or
  - *Edge servers* – recipients connect directly to them
- Scalable bit-stream advantage
  - Transcoders simply truncate & rearrange bit-streams.

Introduction: Scalable Media Infrastructures

- Transcoders often part of a larger infrastructure
  - Media moves down the transcoding chain while capabilities & preferences move up.

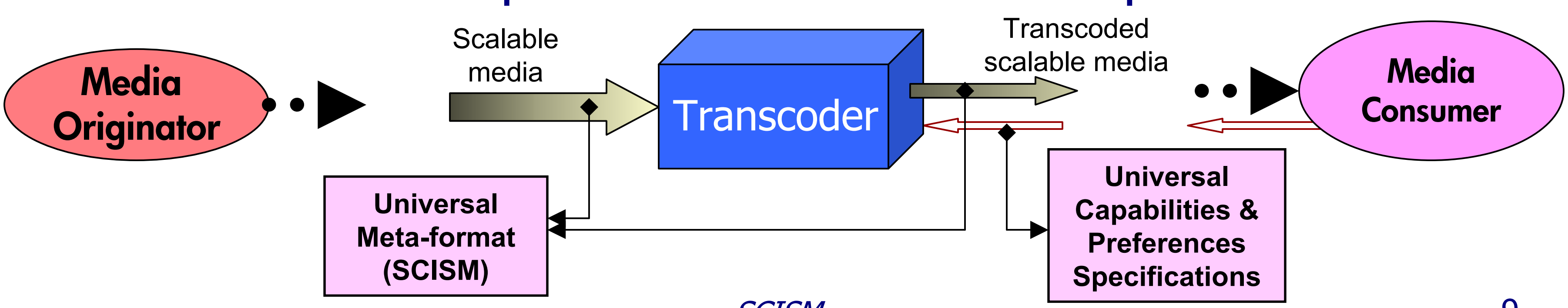


Introduction: Obstacle in Adoption

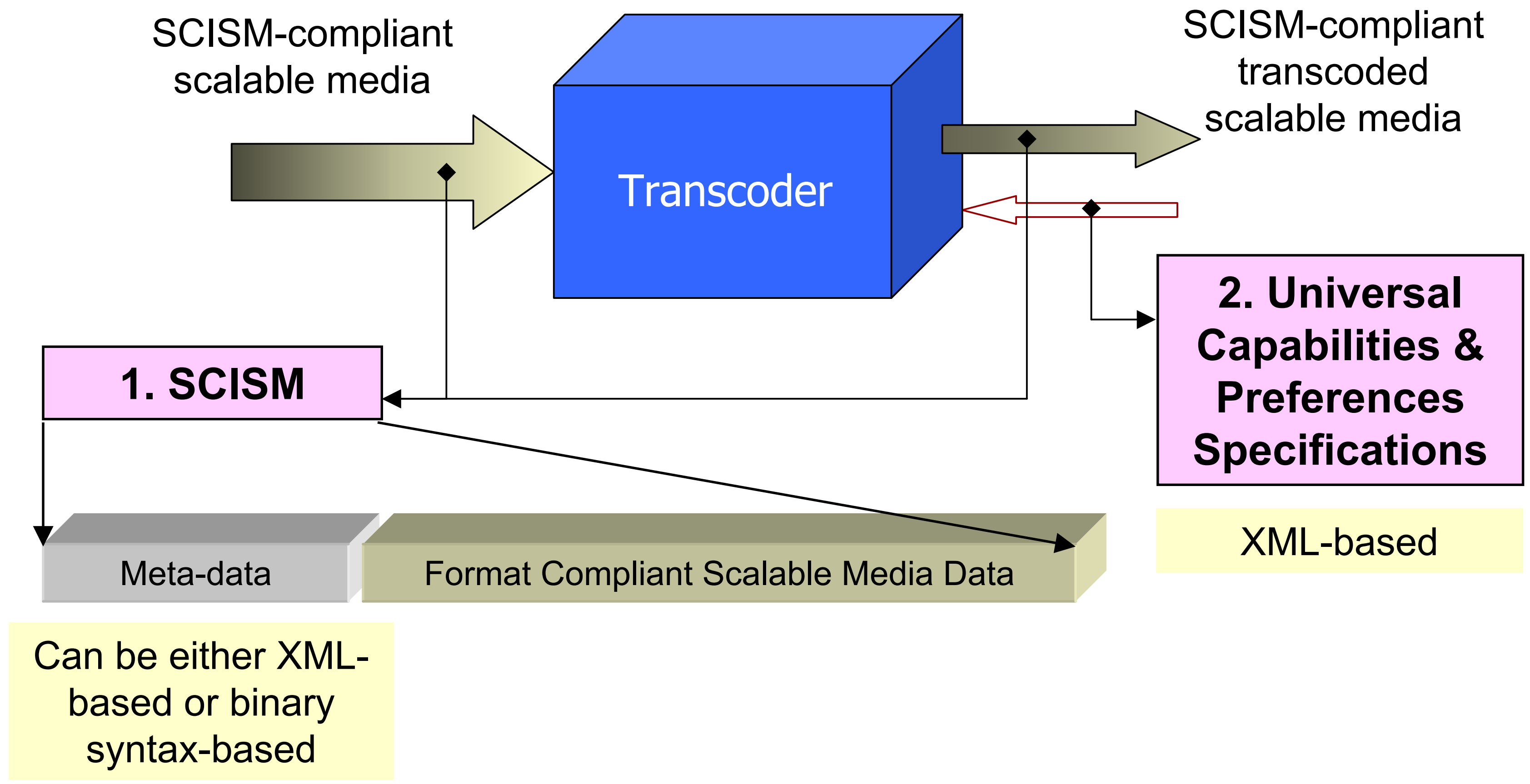
- Transcoders not media-independent
  - They need to be specially designed:
    - for the structure of a specific scalable media-type.
    - to interpret client capabilities & preferences relevant to a specific media-type.
  - Different transcoding infrastructures needed for different types of media
  - Infrastructure **too expensive** to deploy.
    - Standards provide little security, time-consuming.
    - Standards limit flexibility.
- Adoption of Scalable Media need universality:
  - Same infrastructure to work with all media types

Introduction: Objectives of this work

- To develop a framework for universal scalability, leading to evolution of *media-type agnostic* delivery infrastructures.
- Enables seamless ease of multimedia use.
- Need to standardize:
  - The media-format & meta-data [not encoding]
  - The Capabilities & Preferences Specifications



Introduction: Specific Objectives

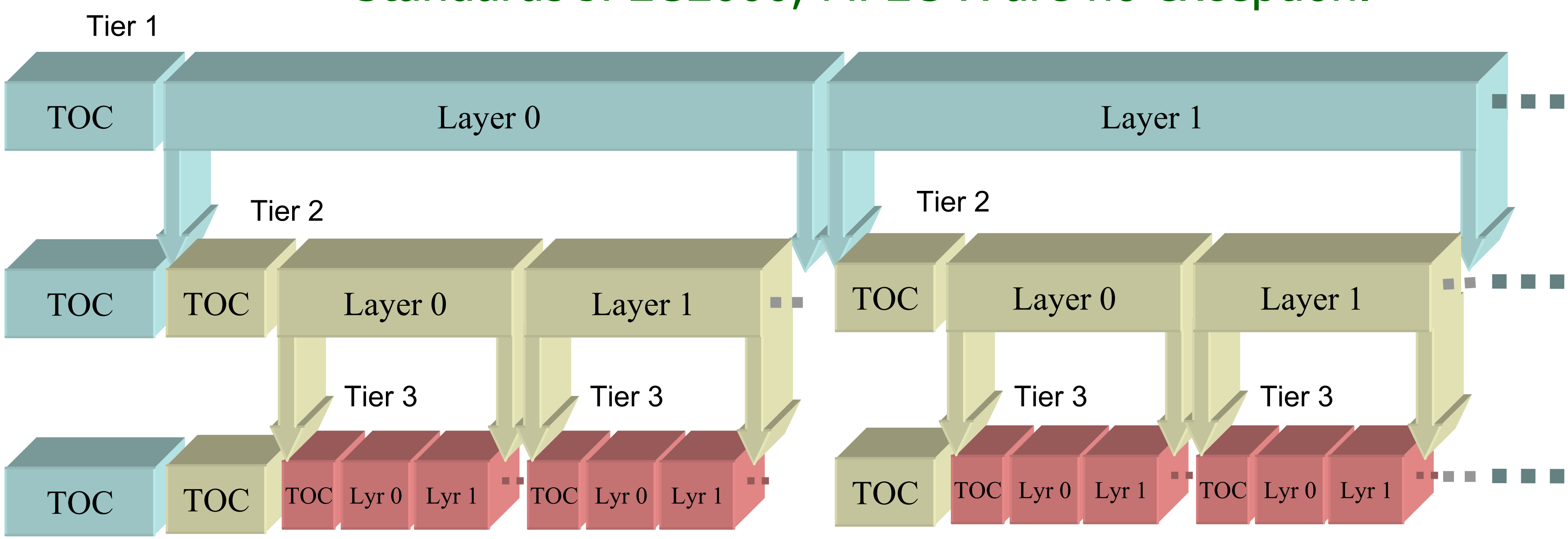


Scalable Bit-streams: Properties

- To create a meta-format we need to:
  - Understand scalable bit-streams.
  - Capture the common characteristics of a scalable bit-stream in a loosely defined *bit-stream format.*
  - Add appropriate *meta-data* in headers describing the specifics for a given media.

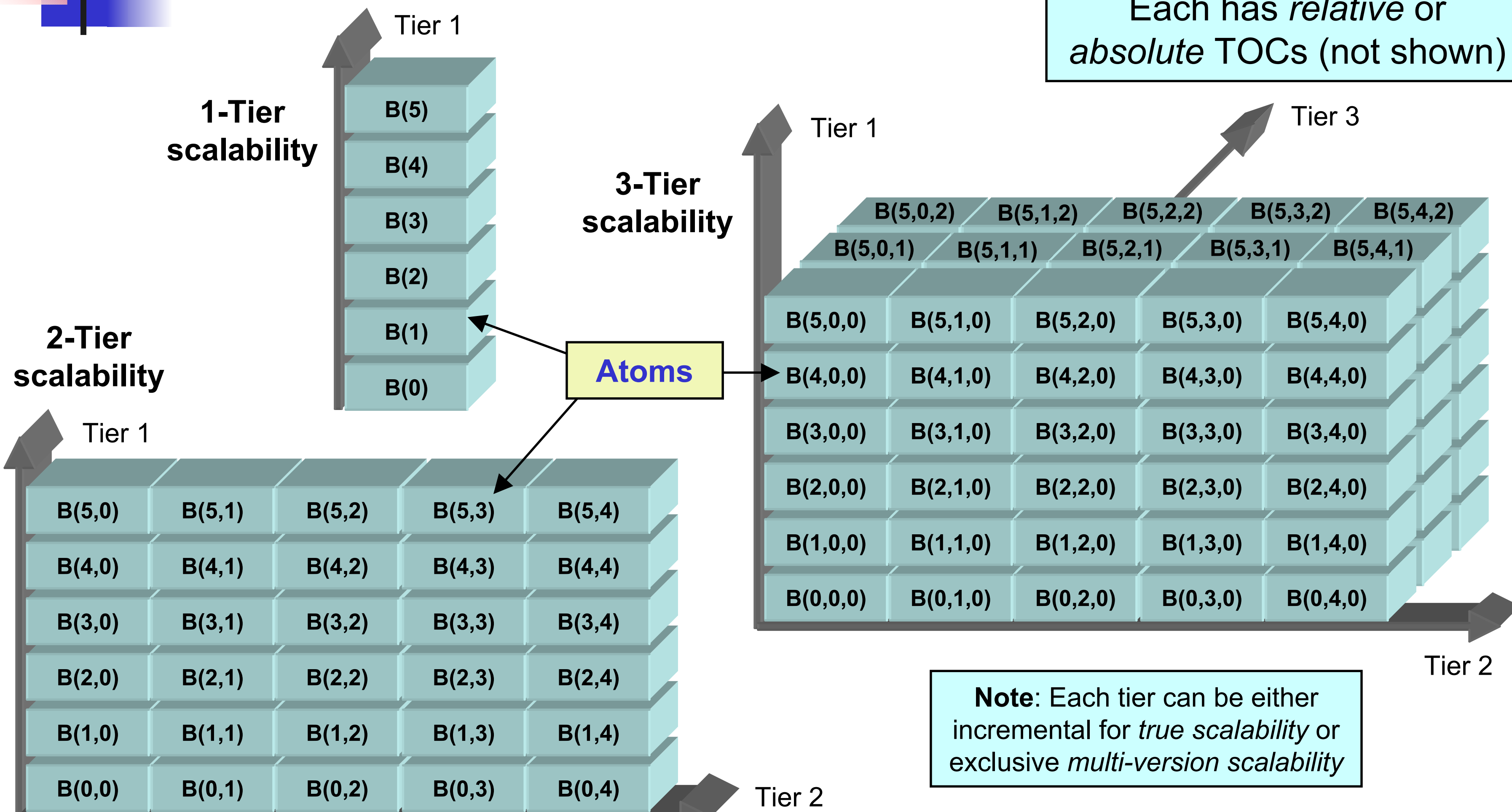
Scalable Bit-streams: Nested Scalability

- Multi-Tier Scalability
  - Scalable bit-streams inherently contain a nested scalability structure
    - Standards JPEG2000, MPEG-X are no exception.



Scalable Bit-streams: Multi-Tier Scalability

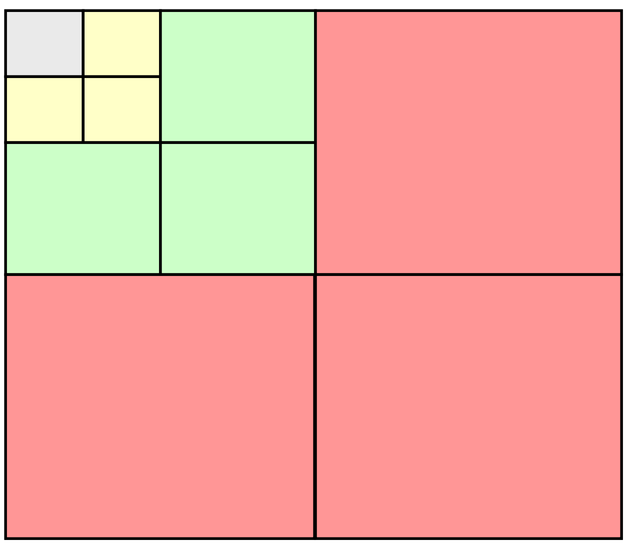
Each has *relative* or *absolute* TOCs (not shown)



Note: Each tier can be either incremental for *true scalability* or exclusive *multi-version scalability*

Scalable Bit-streams: Example

- Image Compression



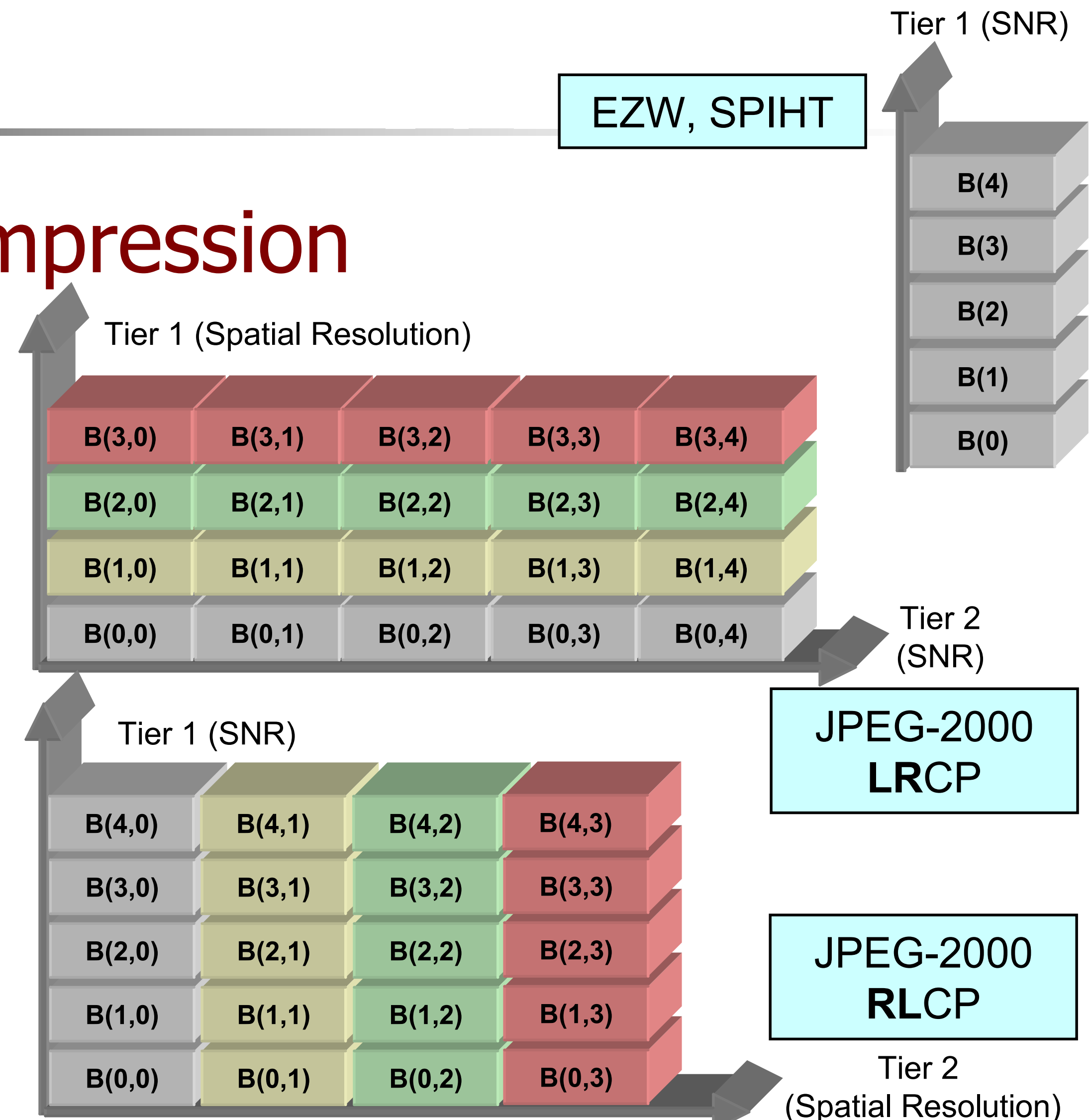


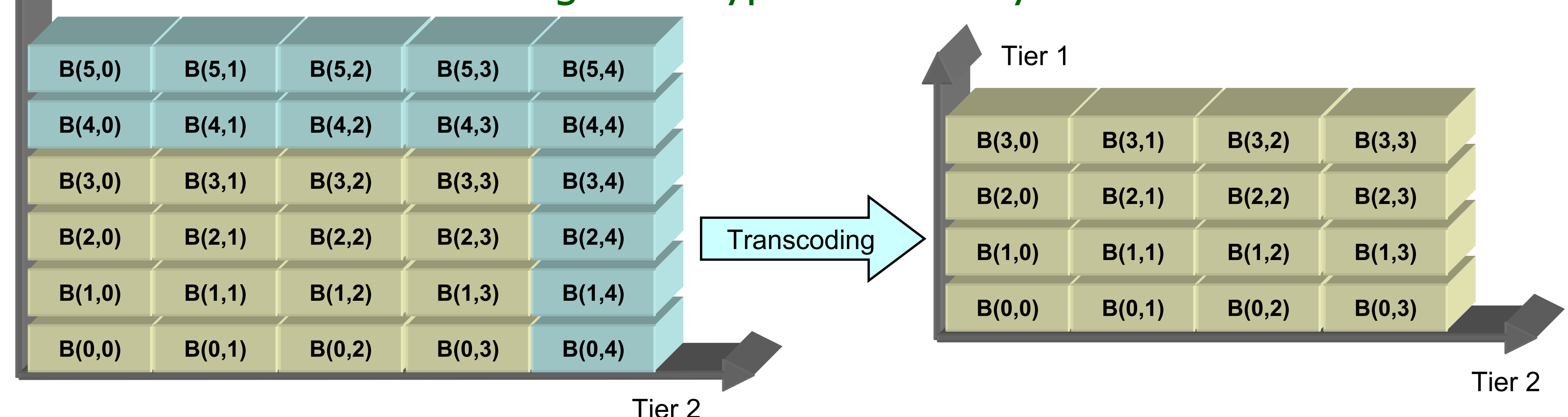
Image Compression
- Wavelet decomposition
- Subband-by-subband coefficient scan
- Bit-plane by bit-plane encoding

Scalable Bit-streams: Examples

- Video Compression
  - MPEG standards support several types of scalability.
  - Streaming Needs
  - They are hardly being used.
- Need to cater to new scalable media
  - Need ways to deliver

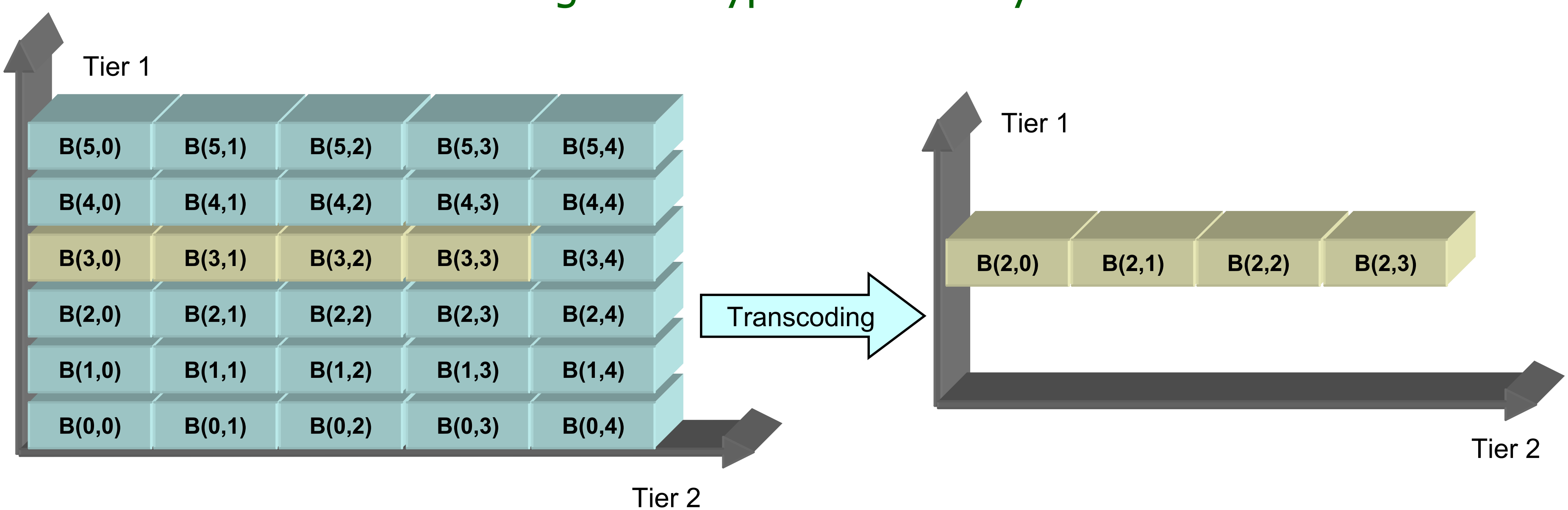
Scalable Bit-streams: Transcoding

- Transcoding comprises:
  - Removing rows and columns of the multilevel bit-stream from outer ends, or selecting single rows and columns
  - No need to *decode* or *decrypt*
    - Encoding & encryption causality must be enforced.



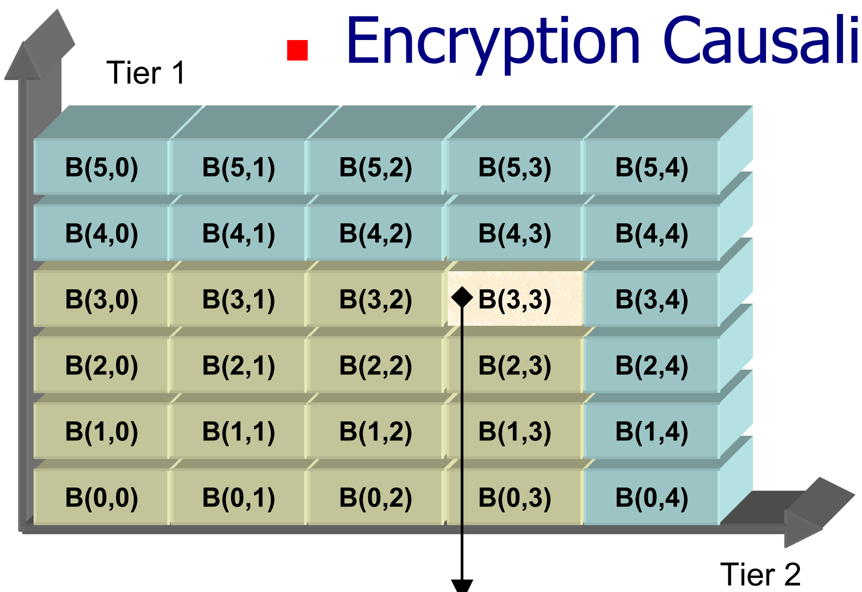
Scalable Bit-streams: Transcoding

- *Exclusive* tiers :
  - Only one layer can be selected in *exclusive* tiers.
  - No need to *decode* or *decrypt*
    - Encoding & encryption causality must be enforced.

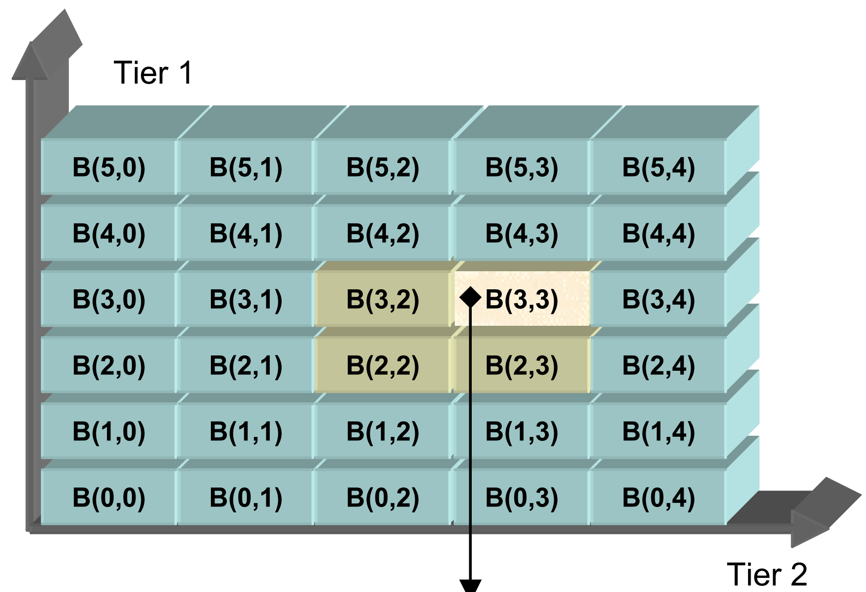


Scalable Bit-streams: Encoding & Encryption

- To make such transcoding work:
  - Encoding Causality
  - Encryption Causality



While *Encoding* this atom, do not use any information from non-causal atoms in incremental tiers, even if they appear earlier in the bit-stream.

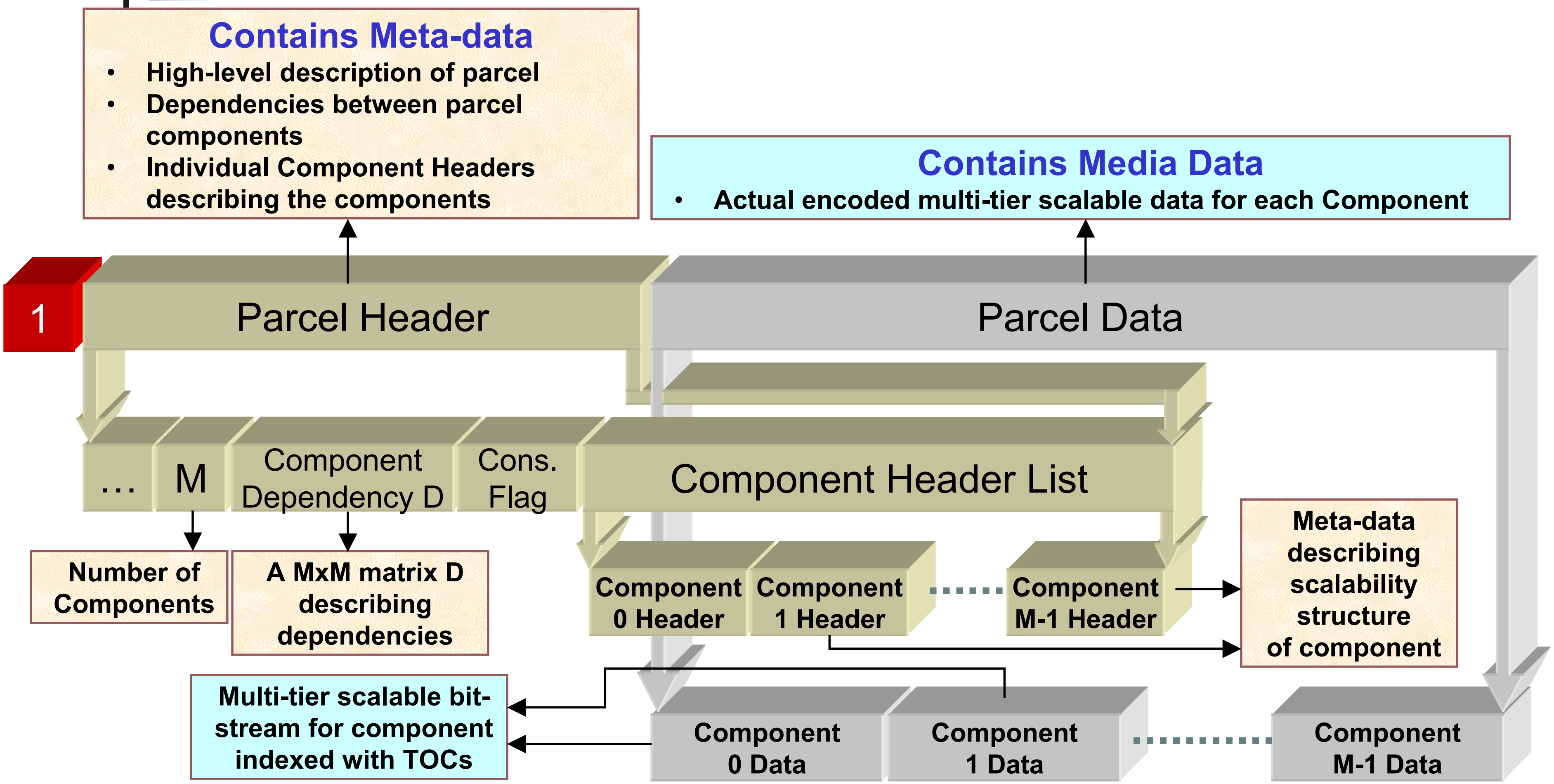


While *Encrypting* this atom, starting state of the generator is derived from the ending states of adjacent causal atoms in incremental tiers.

SCISM: Parcels & Components

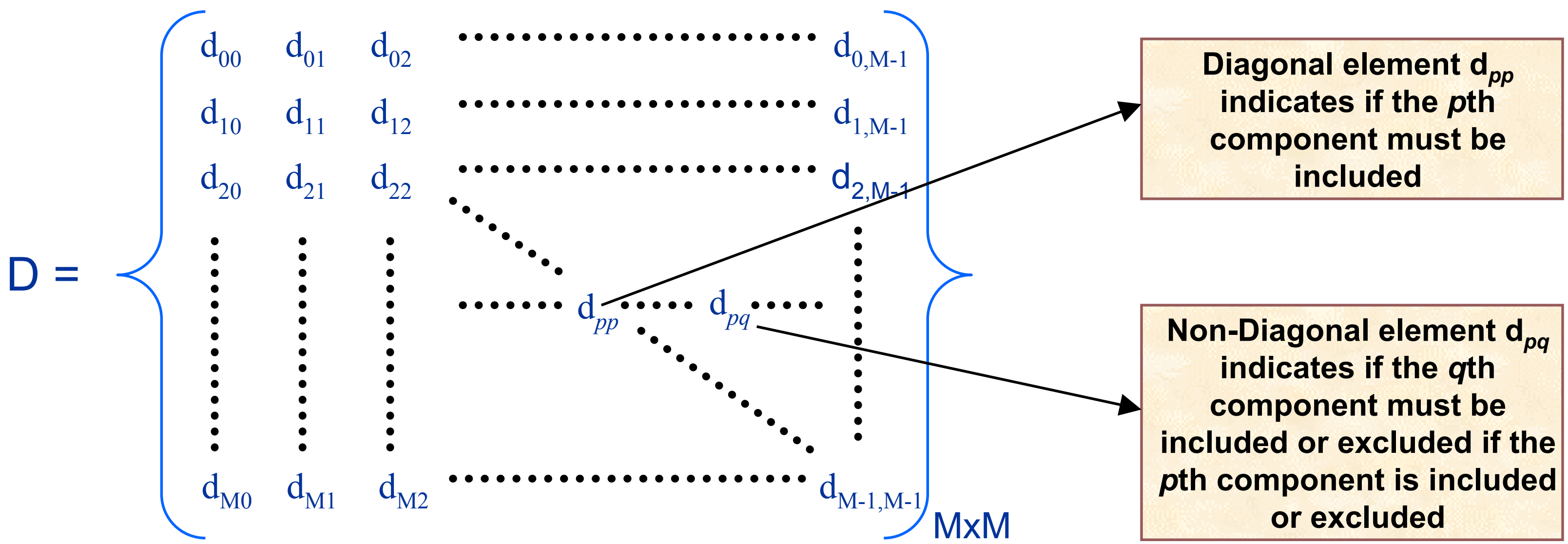
- Media is organized in “Parcels”
  - Parcel is defined as the basic unit of transcoding.
  - Parcelization granularity is a design choice
    - Not an intrinsic property of the media
  - Parcel can have multiple “Components”
    - Components define encoding boundaries that are organized in parcels.
- Example:
  - Parcel – An image with audio commentary.
    - Components – Image data, Audio data

SCISM: Top Level Parcel Format

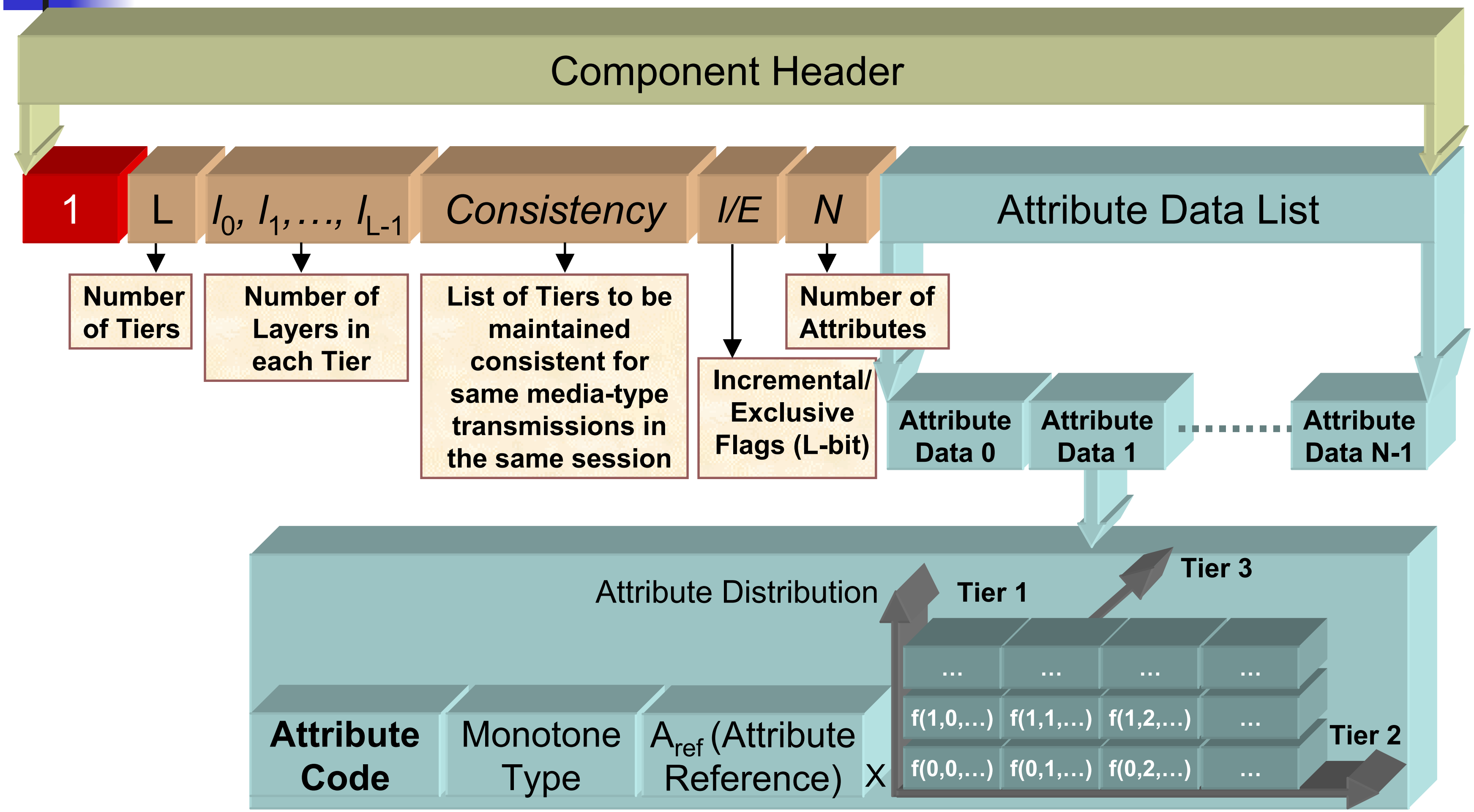


SCISM: Component Dependency

- Component Dependency
  - A MxM matrix D defines dependencies between Parcel components.



SCISM: Component Header



SCISM: Attributes – A Crucial Bridge

- Component attributes play a crucial role:
  - They bridge the communication gap between:
    - The media Creator/Originator (Media meta-data) and,
    - The media Consumer (Capabilities & Preferences Specs).
  - Attributes are:
    - To the **Media Creator**, quantifiable properties of media based on which one can transcode.
    - To the **Media Consumer**, quantifiable properties to indicate its limitations and preferences.
    - To the **Transcoder**, just numbers, based on which it must determine how to transcode a scalable bit-stream.
  - Attributes are quantified by non-negative values called Attribute Values.

Attribute Examples

- Size
- Display Resolution
- Processing Power
- Number of Speakers
- Number of joystick buttons

SCISM: Attribute Codes

- Each attribute is identified uniquely by a code (32-64 bits long)
  - Types of Attributes
    - **Reserved**: Universal across media-types (Size, Display_Resolution etc.)
    - **Custom**: Companies buy *Attribute Code Space* to denote properties of custom media they create
  - Designated field in Attribute Code indicates how to combine attribute values for 2 or more media components.
    - Additive, Maximum, Minimum, Multiplicative etc.

Attribute Combinations

For media with several components, transcoder can jointly optimize with respect to attributes common to two or more components such as size.

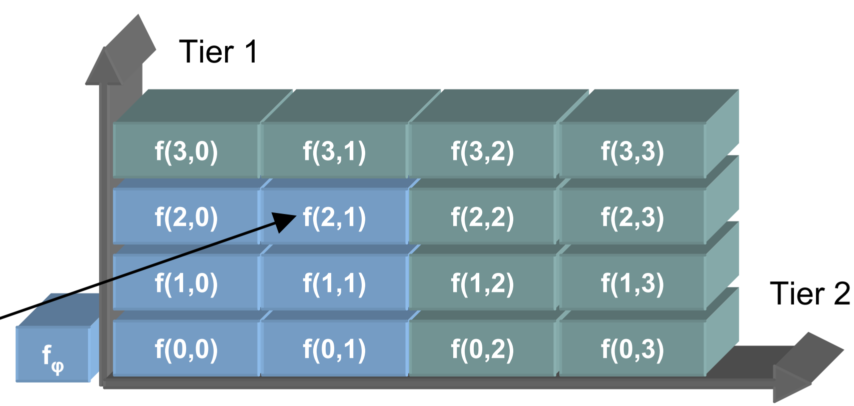
SCISM: Component Attribute Data

Attribute Data

- **Attribute Code**
- **Monotone Type**
  - Non-increasing
  - Non-decreasing
  - Non-monotonic
- **A_{ref} (Reference Attribute Value)**
- **Attribute Distribution over atoms**
  - Multiplier of A_{ref} to yield Attribute value for all transcoding operations. Specified 'exactly' or with 'marginals'

If $A(i,j,\ldots)$ is attribute value for selecting atoms up to $(i,j,\ldots)$, $A(i,j,\ldots) = A_{ref} . f(i,j,\ldots)$

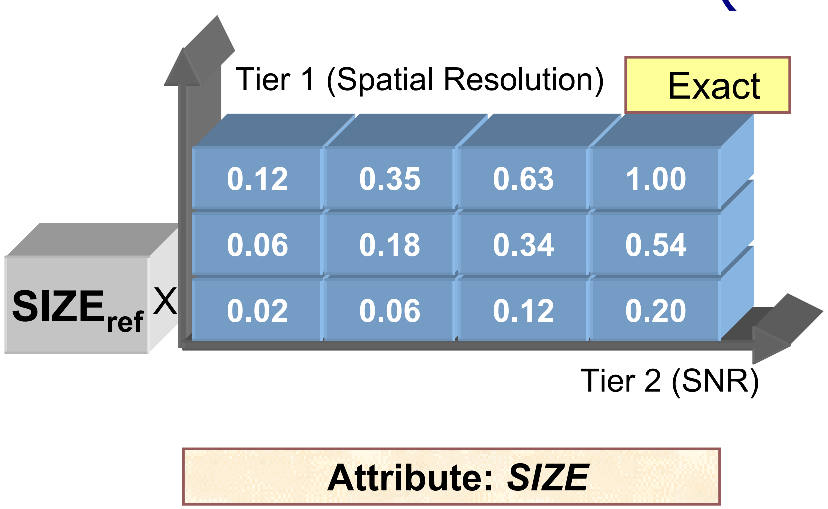
If A_{φ} is default attribute value for dropping the component $A_{\varphi} = A_{ref} . f_{\varphi}$

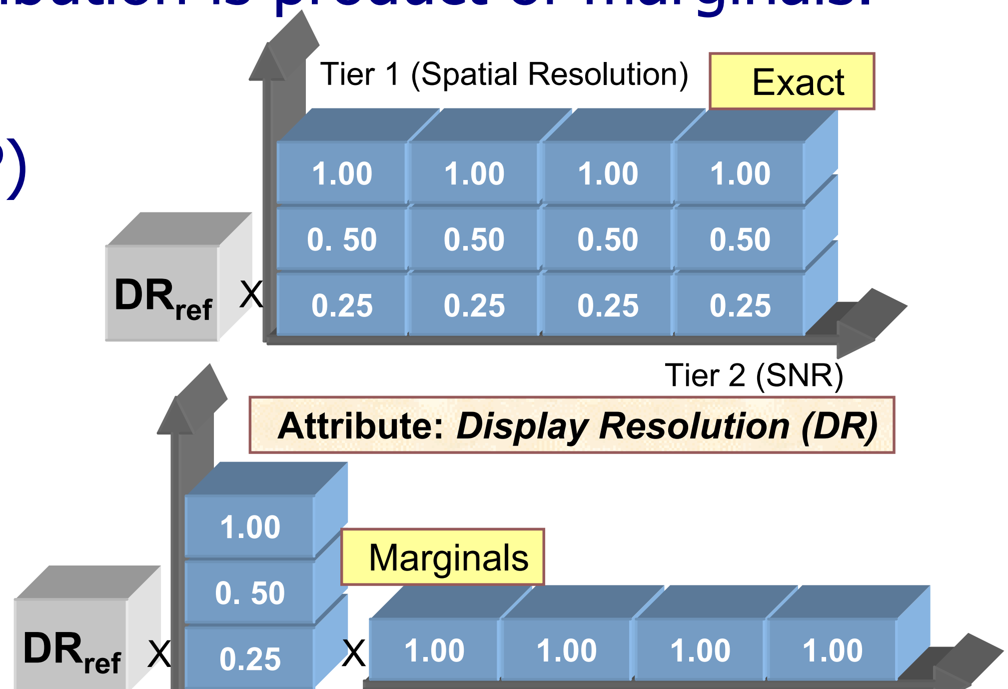


Ex. $A(2,1)=A_{ref}.f(2,1)$

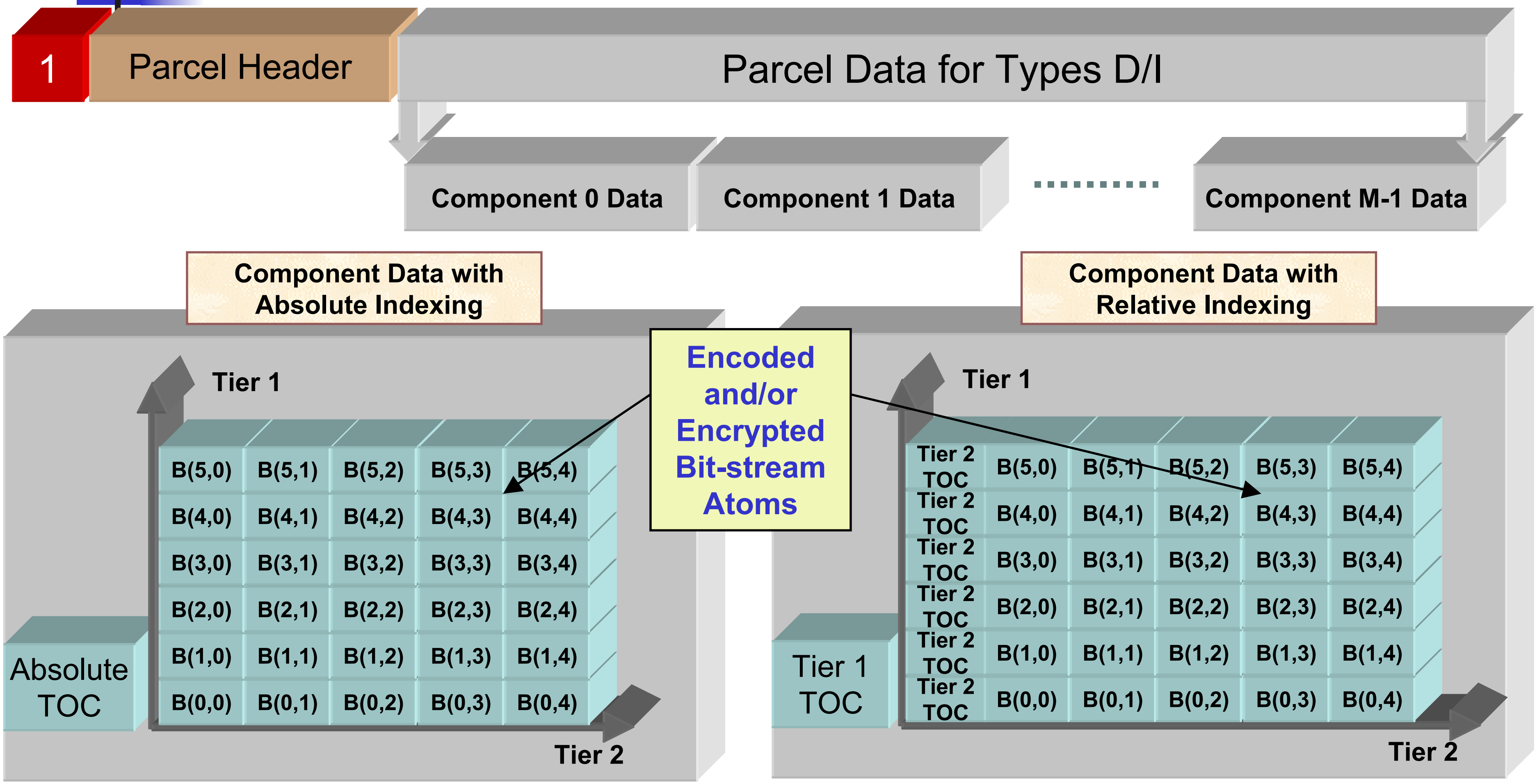
SCISM: Attribute Distributions

- Distribution Specifications
  - Exact
  - Marginals – Distribution is product of marginals.
- Example
  - JPEG2000 (**LR**CP)





SCISM: Parcel Data Format



Capabilities & Preferences: Use of Attributes

- Clients indicate their capabilities and preferences using attributes.
  - Reserved attributes are standardized
  - Custom attribute codes known to sw/hw system used for experiencing the media
- Transcoder needs to interpret the capabilities unambiguously.

Capabilities & Preferences: Constraint Definition Language

- Clients specify constraints on definable functions of attributes, called *Measures*:
  - Measure Definition:

$$M(\bar{a}) = \sum_{i=1}^{N} \lambda_i \prod_{j=1}^{n_i} f_{ij}(a_{ij})$$

N = Number of product terms
n_i = Number of elements in ith product
λ_i = Multiplier for ith product
$f_{ij}(x)$ = Simple univariate functions like $x, 1/x, \log(x), e^x$ etc. with codes standardized in the protocol
a_{ij} = Attribute codes

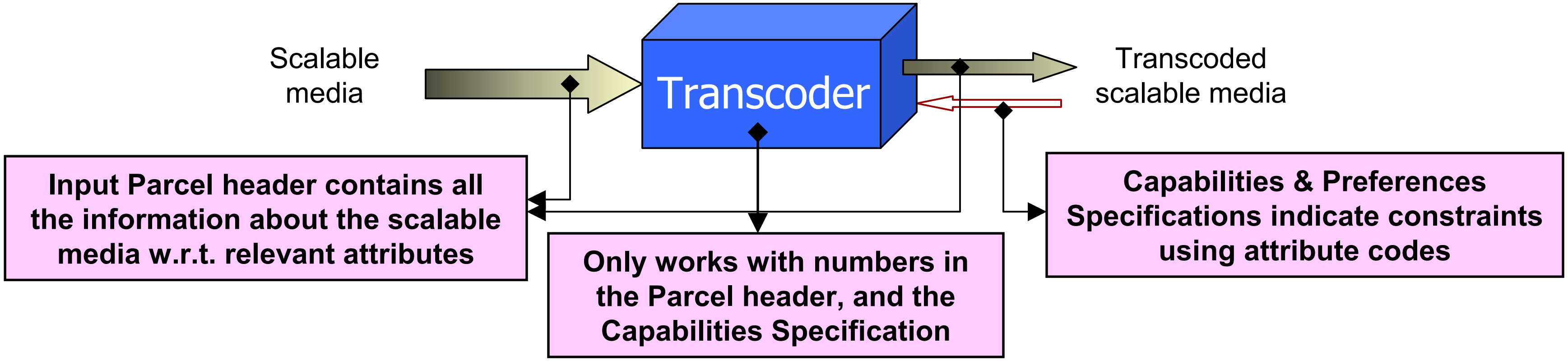
Constraint Examples

SIZE/T < 300
DR > 1280
Min(MSE + λ.SIZE)

  - Constraints:
    - Limit Constraints – *Upper limit, Lower limit, Range* of measure values
    - Optimization Constraints – *Maximize, Minimize* measures

Transcoding Operation: Meta-data & Constraints

- Transcoder gets all the information it needs to decide which layers to drop from:
  - Meta-data in input Parcel header
  - Capabilities & Preferences Specifications
- Transcoder decides layer drops based on ‘numbers’ – need fast optimization algorithm



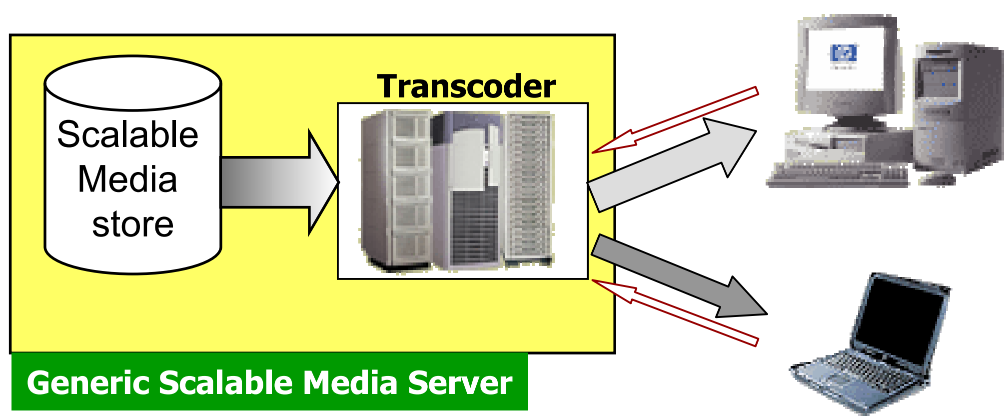
Transcoding Operation: Tasks

- Transcoder acts on a parcel only if constraint measures involve media attributes.
  - Optimization Decisions:
    - Chooses minimal layer drops meeting limit constraints.
    - Within the space satisfying limit constraints, optimizes the optimization measure if any.
    - Maintains Component Dependencies.
    - Maintains layer drop consistency for Tiers in Component Consistency List.
    - Best effort if not all constraints can be satisfied.
    - Monotone type flag aids optimization.
  - Bit-stream manipulation:
    - Drops atoms, repacks and updates TOCs appropriately

Example Scenarios: Scalable Media Server

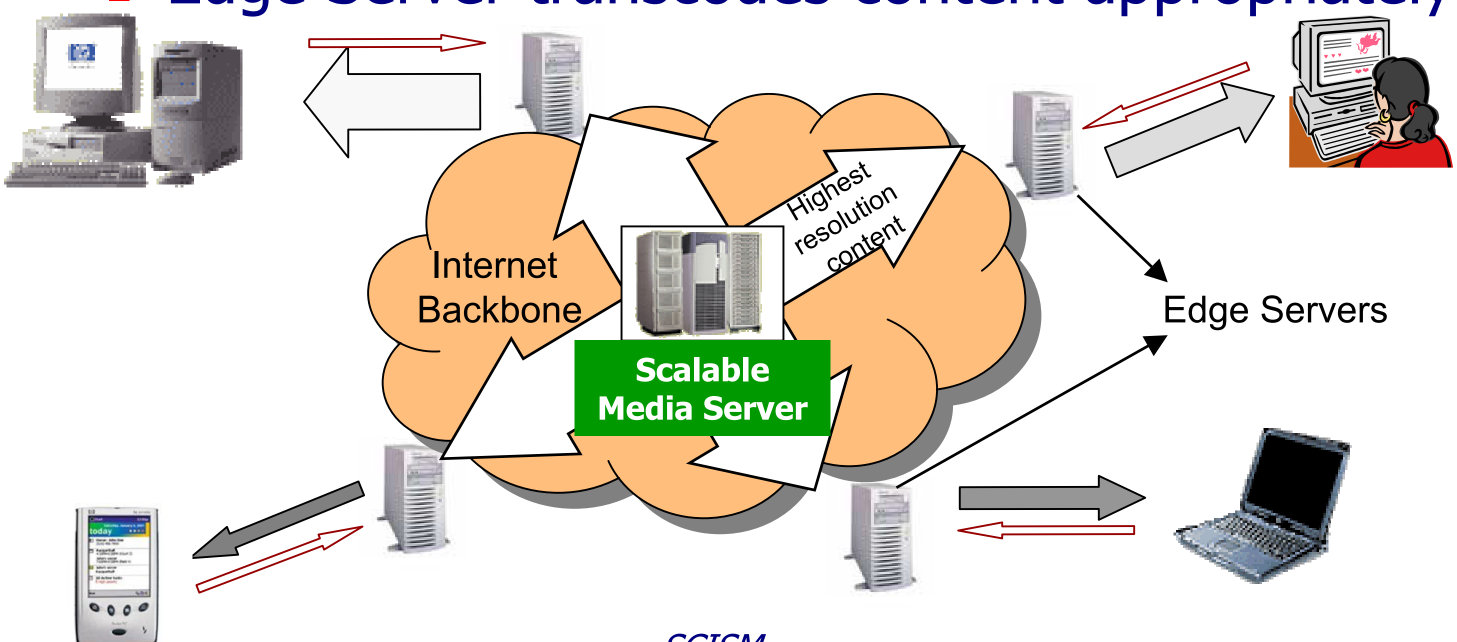
- Simplest scenario – Scalable Media Server:
  - Stores conformant scalable media
  - Accepts client requests with Capabilities & Preferences
  - Transcodes content appropriately and delivers to client directly

Scalable Media Server does not need to know or care what the media-type is.



Example Scenarios: Edge Servers for Transcoding

- Clients connect to Edge Servers to receive content
  - Specifies their Capabilities & Preferences
  - Edge Server transcodes content appropriately



Conclusion: Summary

- A systematic methodology for universal media-type agnostic transcoding of scalable media, has been presented.
  - Bit-stream format and meta-data for scalable media (SCISM) proposed.
    - Encoding standards are less important in this paradigm
    - SCISM is a loose format for all scalable encodings
  - Protocol for Capabilities & Preferences Specifications proposed.
  - Both individual media components, as well as combinations for a overall experience of media, have been considered.

Conclusion: Technology Promises

- Scalable media can revolutionize the world:
  - Seamless content adaptation and ease of use
  - Automatic explosion in use of media
- 'Media' not limited to images/audio/video
  - New scalable media will evolve – 3D, 4D
  - Encoding standards too slow to keep up
- The proposals in this talk promise:
  - To remove the infrastructure expense bottleneck.
  - To make adoption of new scalable media easy by removing reliance on encoding standards.
  - To unlock the potential of scalable media.

Conclusion: Sign Off

- Contact:
  - Debargha Mukherjee
  - Amir Said
  - Giordano Beretta

HP Laboratories
1501 Page Mill Road, MS 1203,
Palo Alto, CA 94304.
Email: {debargha, said, beretta}@hpl.hp.com