

# Introduction to OpenTravel 2.0

## Overview, Benefits and Development Process

**NOTICE:** The content in this presentation is **DRAFT** and is subject to change before the initial 2.0 Publication. It is **NOT** advisable to use any of these materials from an implementation planning perspective.

# Topics

2

- What is OpenTravel 2.0?
- Business Benefits
- Member & Implementer Benefits
- 2.0 Development Process

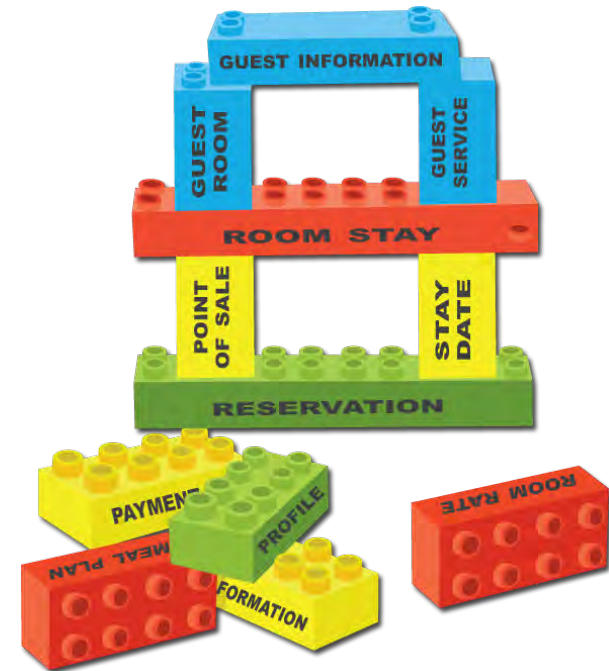
# What is OpenTravel 2?

- introduction
- why is opentravel 2 necessary?
- key differences between opentravel 1 and 2
- future plans for opentravel 1 and 2

# What is OpenTravel 2? (1/2)

4

OpenTravel 2 is an **xml component suite**—based on a common travel industry vocabulary—that allows trading partners to construct xml messages.



# What is OpenTravel 2? (2/2)

5

- OpenTravel 2 addresses the changing way businesses use XML in the travel marketplace by providing:
  - Reduced transaction sizes from lighter schema
    - Allowing implementers to use only what they and their trading partners need
  - Easier integration with non-OpenTravel XML
    - Reduced binding issues and namespace complexity
  - Faster internal application development
    - From increased modularity and more frequent releases
  - Consistent trading partner implementations
    - From a more precise specification, with less ambiguity & greater extensibility
  - Easy integration of proprietary business logic
    - Multiple extension points for proprietary business logic and functionality

# Key Differences Between 1.0 and 2.0

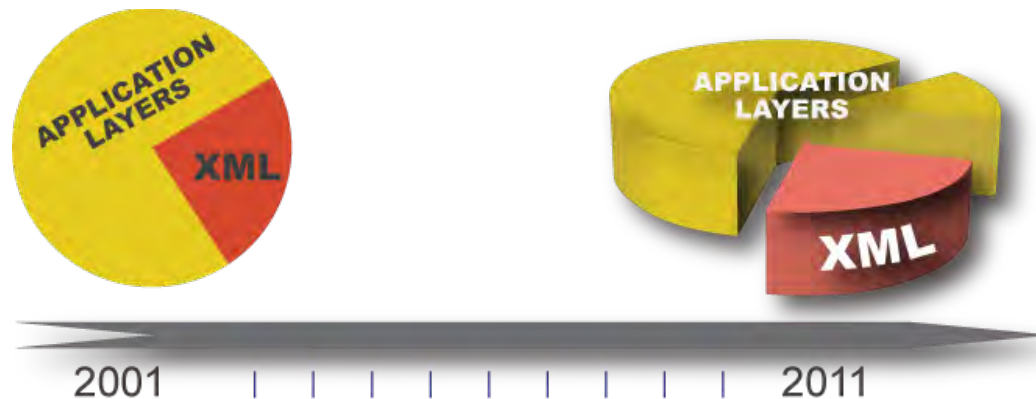
6

	OPENTRAVEL 1.0	OPENTRAVEL 2.0
<b>SPECIFICATION</b>		
Architecture	<ul style="list-style-type: none"><li>• XML Messages</li></ul>	<ul style="list-style-type: none"><li>• Business Objects in component sets</li><li>• XML reference implementation messages &amp; WSDL</li></ul>
<b>PROCESS</b>		
Release Cycle	<ul style="list-style-type: none"><li>• 6 month publication cycle</li><li>• No iterative release</li></ul>	<ul style="list-style-type: none"><li>• 6 month publication cycle</li><li>• Iterative release in 30 days</li></ul>
<b>TOOLS</b>		
	<ul style="list-style-type: none"><li>• Message Users Guide</li><li>• Schema Dictionaries</li></ul>	<ul style="list-style-type: none"><li>• Web-based, searchable XML Component Dictionaries</li><li>• Reference implementation schema</li><li>• Implementer-tailorable Usage Profiles</li></ul>

# Why is OpenTravel 2 Necessary? (1/3)

7

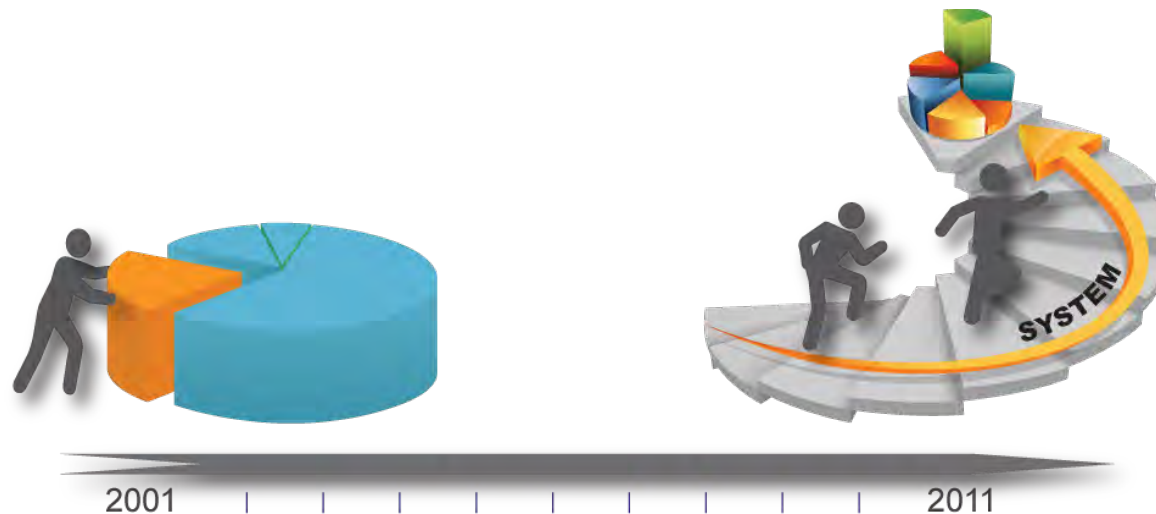
- The role of XML standards has shifted...
  - **From** being core functionality embedded in monolithic applications
  - **To** being a modular reusable library



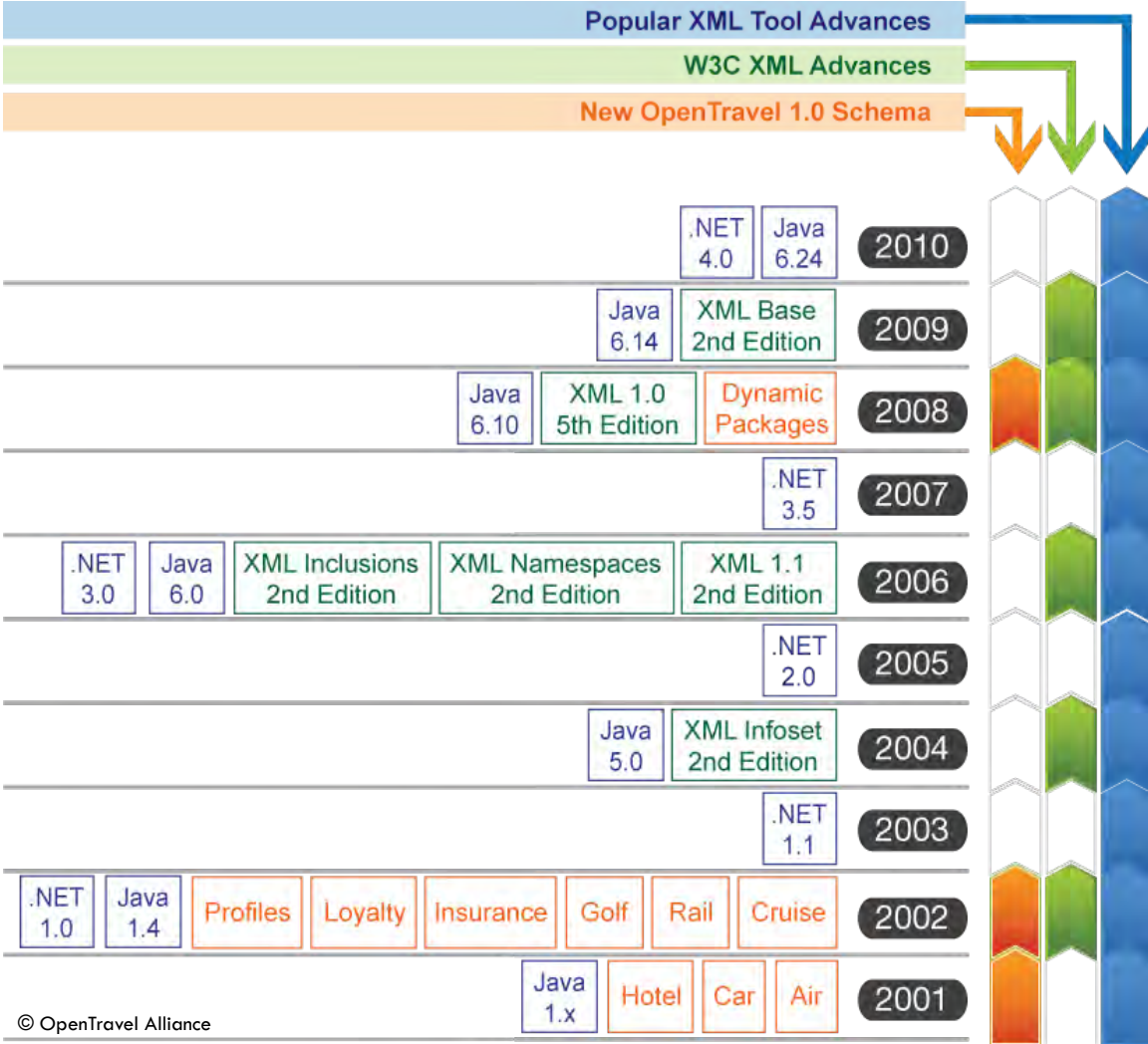
# Why is OpenTravel 2 Necessary? (2/3)

8

- Typical XML consumption has changed...
  - **From people** reading schemas and building monolithic applications
  - **To systems** reading schemas and building reusable classes from which applications are built



# Why is OpenTravel 2 Necessary? (3/3)

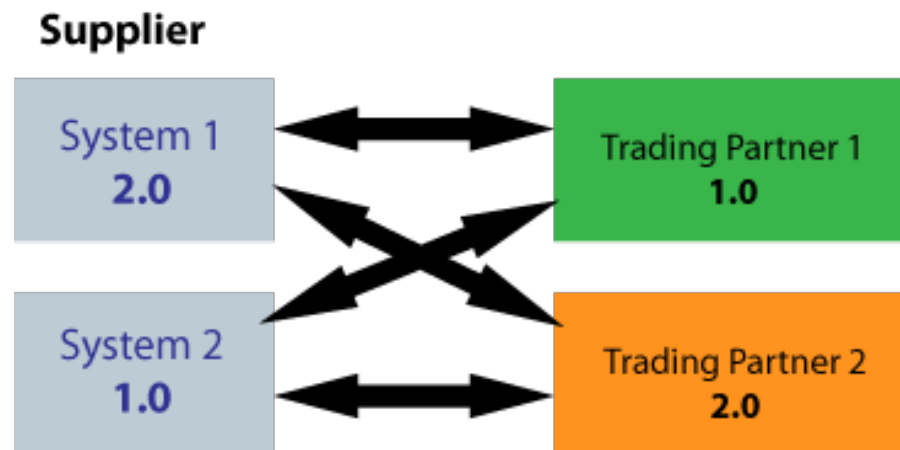


- OpenTravel schema design has not matured at the same pace as WC3 XML and popular XML development tools

# What About OpenTravel 1.0?

10

- OpenTravel 1.0 is not being sunsetted
  - OpenTravel 1.0 will be supported/ enhanced for the foreseeable future
  - Requested enhancements to OpenTravel 2 will be synchronized with 1.0 (and vice-versa)
- OpenTravel 1.0 enabled systems can “speak” to OpenTravel 2.0 enabled systems (and vice-versa)



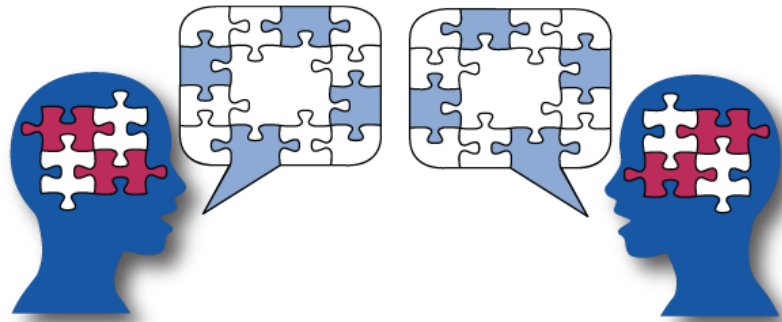
# Business Benefits

- consistent vocabulary
- proprietary business logic support
- faster time to market
- happy IT managers
- tailorable tools & supporting resources

# 2.0 Business Benefits (1/5)

12

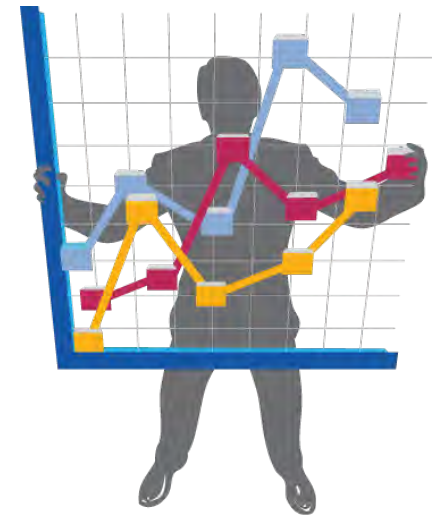
- **A consistent vocabulary for the travel industry**
  - If everyone is speaking the same “language” business benefits from:
    - Faster internal system design and development
    - Faster trading partner implementations



# 2.0 Business Benefits (2/5)

13

- **Helps build competitive advantage by supporting proprietary business logic**
  - Provides extensible and componentized XML business objects that:
    - Allows implementers to focus on proprietary business logic/ processes
    - Provide unique extensibility to support:
      - Corporate business logic and processes
      - Unique trading partner processing
      - Individual application/ system processing



# 2.0 Business Benefits (3/5)

14

- **Faster time to market**
  - Improved alignment with project development/deployment cycles
  - Faster enhancements and easy extensibility support internal development and deployment schedules



## 2.0 Business Benefits (4/5)

15

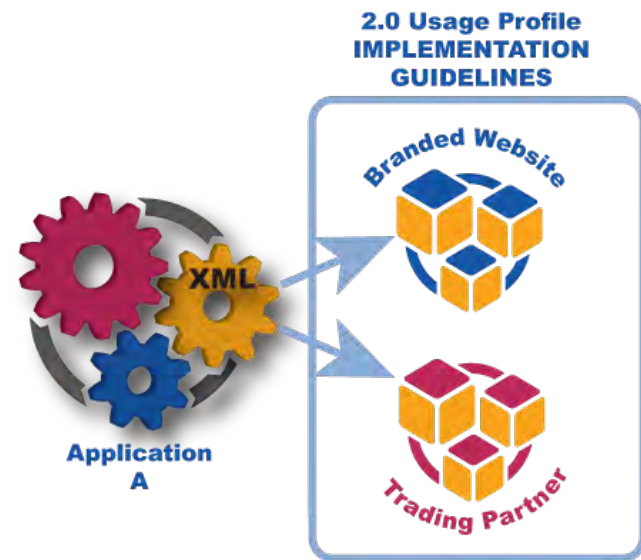
- **Happy developers  
= HAPPY  
MANAGERS**
  - Easy for developers to learn & reuse
  - Better alignment with their new generation of XML and application development tools



# 2.0 Business Benefits (5/5)

16

- **New tools and resources that are application and trading partner friendly**
  - Support learning, development & implementation
    - Resources to that describe 2.0 business objects
    - Resources that describe best practice implementations
    - Tailorable tools to describe general or specific implementations for internal and trading partner use
      - Tailorable by application & platform
      - Tailorable by trading partner



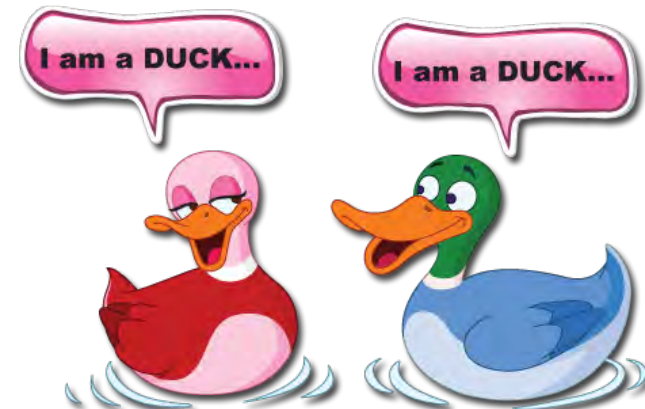
# Implementer Benefits

- standardized xml vocabulary
- easier and faster implementations
- code-oriented schema design
- reduced binding issues
- light-weight schema
- easy extensibility
- faster specification enhancements

# 2.0 Developer Benefits (1/7)

18

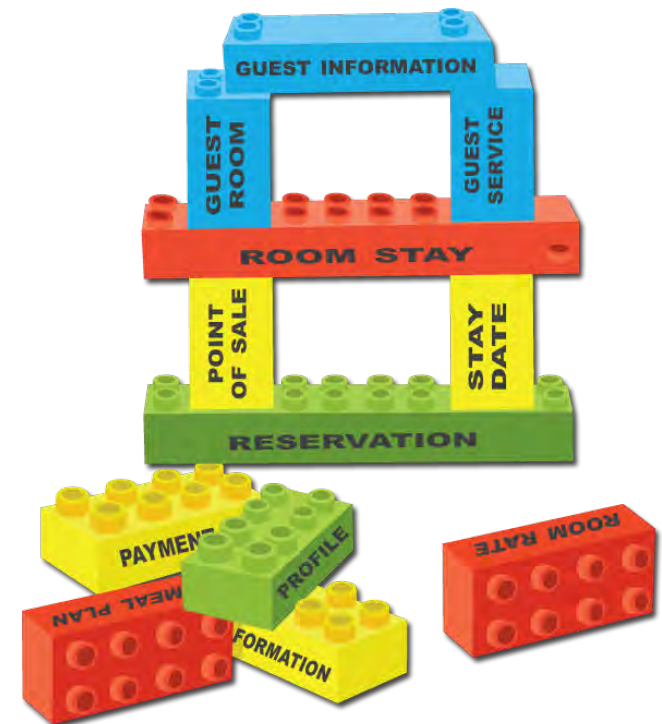
- **Standardized XML vocabulary**
  - Define **once** and **reuse** everywhere
  - Improve design modularity and consistency
    - 2.0 xml components have consistent naming and structural patterns
    - Easy to learn and use



# 2.0 Developer Benefits (2/7)

19

- **Easier & faster implementations**
  - Use only the 2.0 xml component “building blocks” needed for messages
  - Create new business objects from xml component library as needed
  - Refer to (or use) 2.0 “reference implementation” schema library based on industry best practices

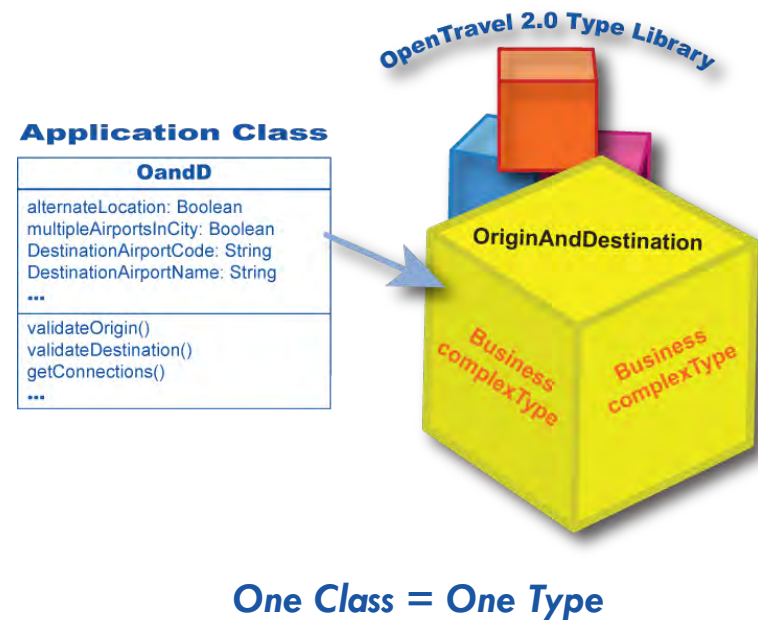


# 2.0 Developer Benefits (3/7)

20

## □ Code-oriented schema design

- Increased tool friendliness
  - Support for web services & other modern messaging frameworks
- Easier OpenTravel schema binding in different tools & languages

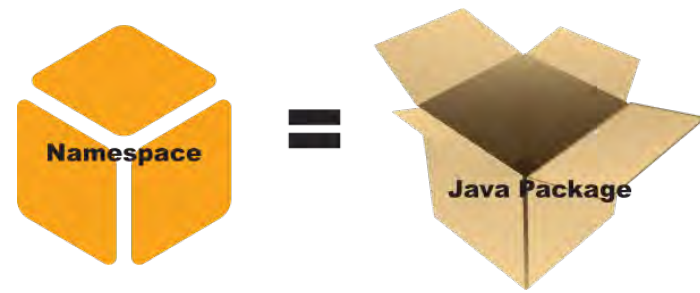


# 2.0 Developer Benefits (4/7)

21

## □ Reduced binding issues

- Minimized namespace intrusion into XML data
  - Allows multiple schemas to be easily be combined into a service
- Unique identification of OpenTravel & non-OpenTravel content
- Removal of nested includes



One Namespace = One Java Package

# 2.0 Developer Benefits (5/7)

22

- **Light-weight schema**
  - Minimized & focused payload
  - Minimized optional elements & attributes
  - Element & attribute de-duplication



# 2.0 Developer Benefits (6/7)

23

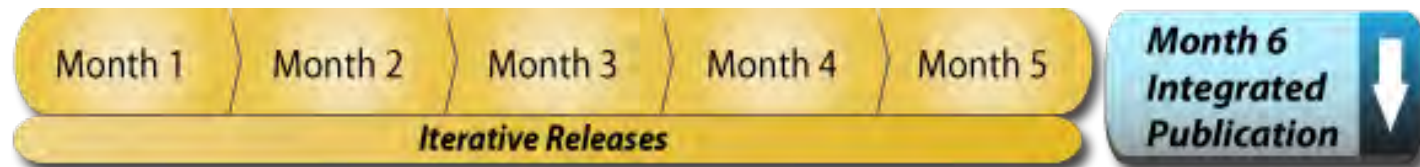
- **Easy extensibility**
  - Extension points can be added to any XML component (or schema message)
  - “Open” enumerated choices
    - Allow extension of enumerated lists
      - No code list or schema enhancement requests



# 2.0 Developer Benefits (7/7)

24

- **Faster xml component library enhancements**
  - Monthly iterative releases
  - Bi-annual (integrated release) publication



2.0

# Development Process

# 2.0 Development Process

26

## **OpenTravel Member Companies Participating in 2.0 Design**

- Amtrak
- ATPCO
- Avis/Budget
- Cybage
- Datalex
- Hewlett Packard
- HTNG
- IBM
- Intercontinental Hotels Group
- Marriott
- Sabre
- Sita Aero
- The Rubicon Group
- Travelport
- Tsinghua University
- TTI

# 2.0 Development Process

27

1. Select OpenTravel vertical (2.0 Team)
2. Start project team
3. Create core & vertical XML components (2.0 team/ project team)
4. Review with OpenTravel workgroup
5. Modify as required (2.0 team/ project team)
6. Create usage profiles and other artifacts
7. Publish OpenTravel XML Type Library
8. Disband project team
9. Repeat step 1...