*An Introduction to ZHENG Qing

by ZHENG Qing

Agenda

- My Profile
- My Study Life
- My Religious Belief
- History of Christianity
- Functional Programming

*Profile

*金牛座 + O型血 + 龙

1. 按部就班，固执已见
2. 按自己的人生哲学走路，不轻易改变自己的生活习惯
3. 从容不迫的实践者，凡事不焦躁，既不想走在别人面前，也不会有依靠别人到达目的地
4. 与其冒着跌倒的危险领先别人，倒不如安全地抵达目的地
5. 有冲动，投机的趋向
6. 有耐心，有毅力；但不善于把握机会，优柔寡断

*Min Hang

*Shanghai


28 Miles
五角场镇
江湾镇
翔殷路
五洲大道
中环路
南翔镇
桃浦镇
彭浦镇
封浜镇
G204
曹路镇
江桥镇
虹口区
杨浦区
普陀区
闸北区
金桥镇
长征镇
东方明珠
广播电视塔
钦洋镇
长宁区
静安区
黄浦区
浦东新区
S222
S121
S308
S206
北翟高架路
新泾镇
卢湾区
世纪公园
张江镇
地铁2号线
唐镇镇
华夏高架路
上海虹桥
国际机场
花木镇
S101
徐汇区
六里镇
北蔡镇
川沙镇
虹桥镇
S202
S102
孙桥镇
S1迎宾高速
机场镇
徐泾镇
G50沪渝高速
七宝镇
磁悬浮
S206
梅陇镇
三林镇
S20外环高速
S224
九亭镇
莘庄镇
康桥镇
闵行区
华泾镇
周浦镇
六灶镇
祝桥镇
泗泾镇
S123
S122
S32申嘉湖高速
盐仓镇
洞泾镇
颛桥镇
坦直镇
三灶镇
S124
S221
新桥镇
下沙镇
惠南镇
马桥镇
吴泾镇
S323
S32申嘉湖高速
航头镇
S122
新场镇
黄路镇
车墩镇
S223
S123
三墩镇
S103
金汇镇
泰日镇
S222
S324
S306
S2沪芦高速
大团镇
西渡镇
齐贤镇
头桥镇
四团镇
叶榭镇
S324

* Age of Empire

http://ageofempiresonline.com/

* Game

Greek

Egypt

Celt

Persia

*Interests

* Movies, TV-Series
  - USA
* Documentary
  - BBC

MARION COTILLARD
MATT DAMON
LAURENCE FISHBURNE
JUDE LAW
GWYNETH PALTROW
KATE WINSLET

NOTHING SPREADS LIKE FEAR

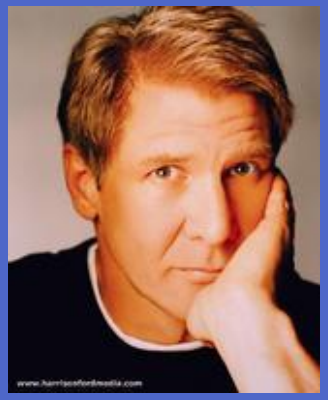
CONTAGION


www.mtime.com
FROM THE DIRECTOR OF IRON MAN
COWBOYS
&
ALIENS
IN THEATERS JULY 29
时光网
mtime
www.mtime.com



*Starts






*Starts

Career Goals

- College Professor
- Corporation Scientist

*Study Life

*Undergraduate

*2006 ~ 2010

School of Software, SJTU

Junior: Fundamentals of Software Development

Junior: Enterprise System Architecture

Senior: Software System Architecture

*Research History

*Workflow Verification

WYS

*Cloud Resource Management

CX, WWT, LGD

*Research Direction

* Cloud Storage
  - Management of Data
  - Distributed Computing
  - Cloud Computing

  ZC (Sister Volcano)

*Religious Belief

* Christianity
  * Catholic
  * Orthodox
  * Protestant

* God

* Determined
  * God has plan for the world and every one of us
  * We have already made all the decisions
* Example
  * We would still make the same decision as we have made if time could go back and we had to do that chose again
  * No matter how hard the chose is or how much time we spent on evaluation, we are bound to make the pre-determined decision

* View of the World

- Free
  - We are free to make our own chose as we see fit
  - We should be responsible for every decision we make and therefore our future
- Example
  - No one can help or force us to make any decision
  - What we decide right now will certainly make our future different

View of the World

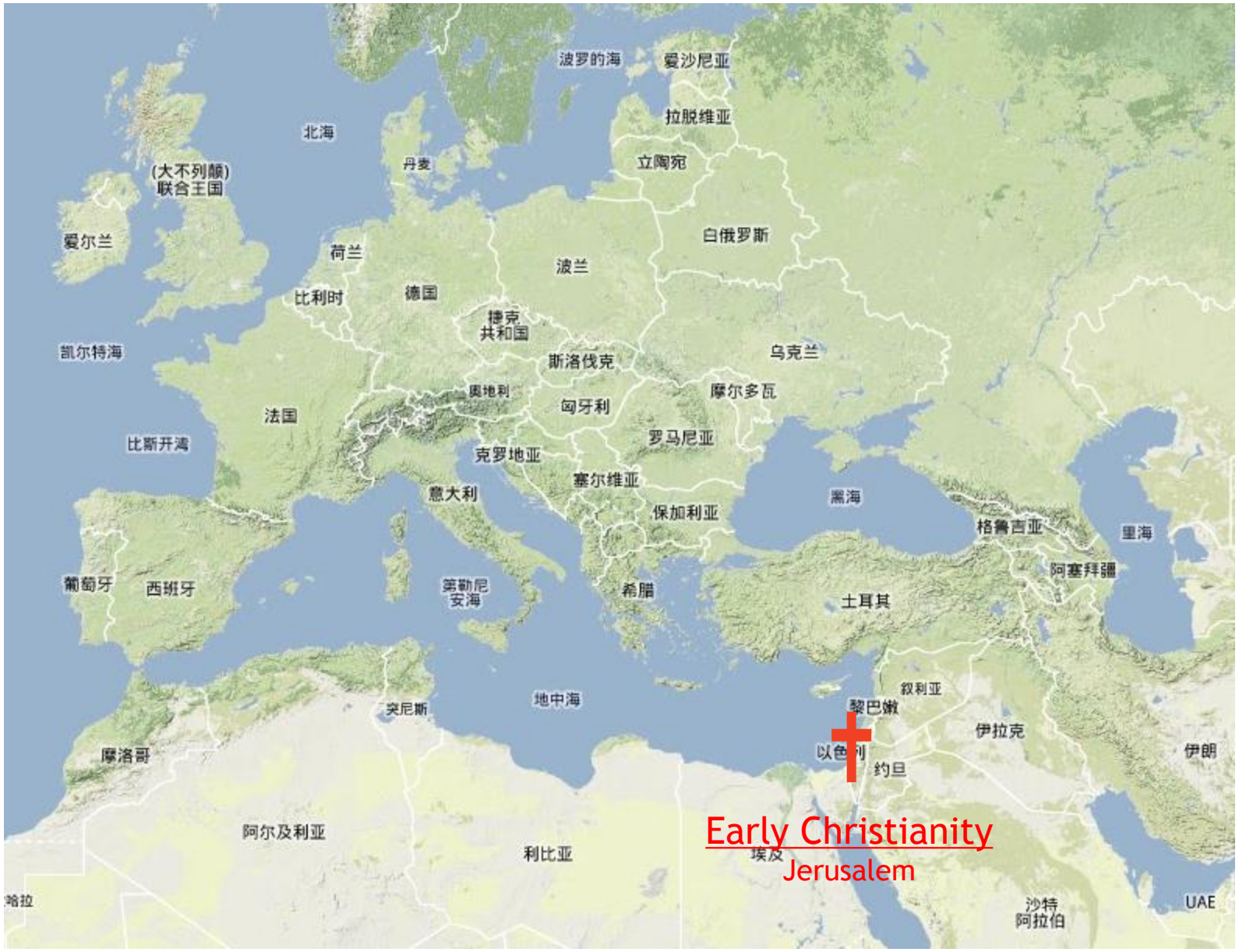
* Summary
  * The Path of Life is determined in the view of God; Every thing happens for a reason
  * Life is unpredictable and full of surprise as we see it, so make every decision carefully and wisely
  * Do not regret for the decisions we have already made; There is always hope!

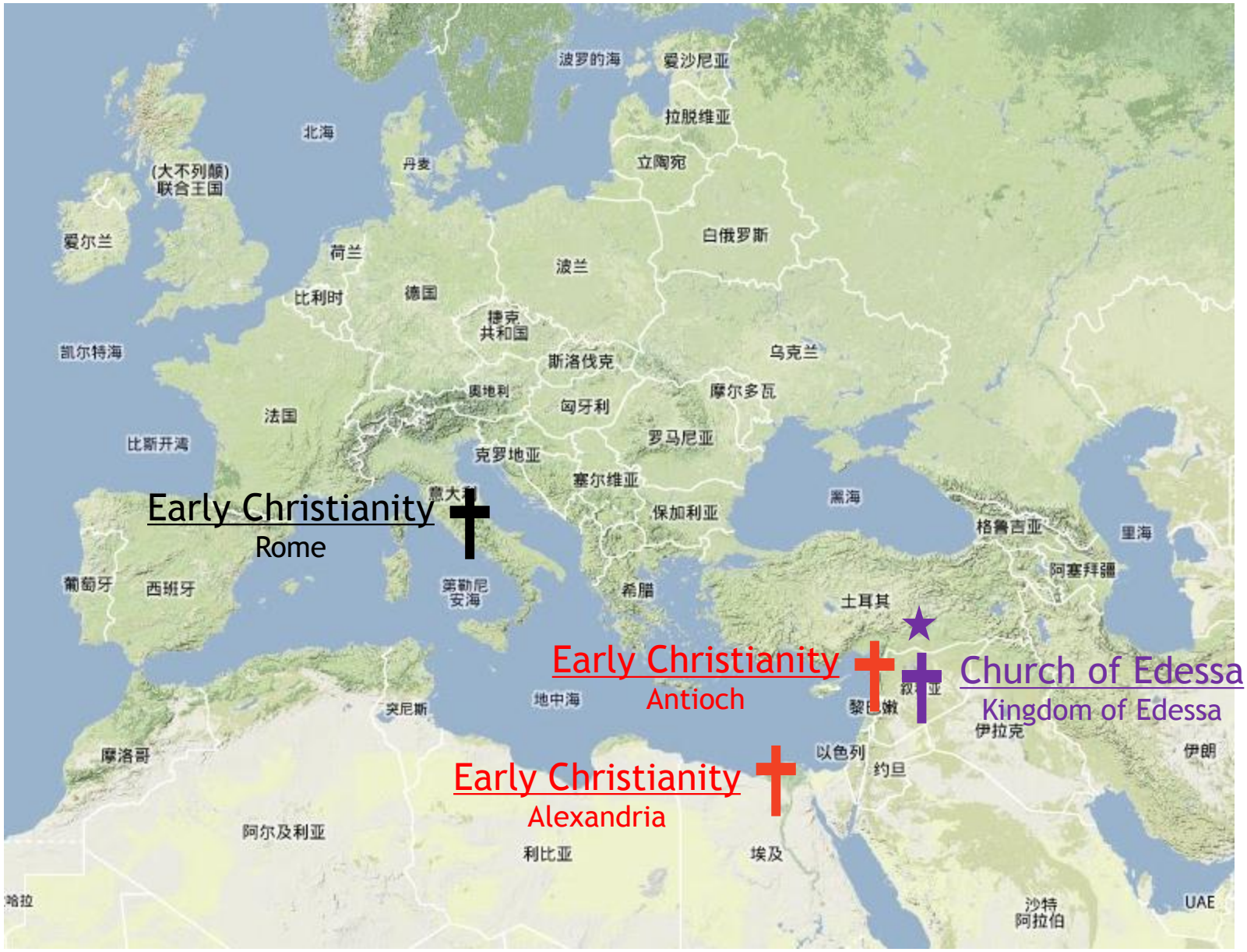
* View of the World

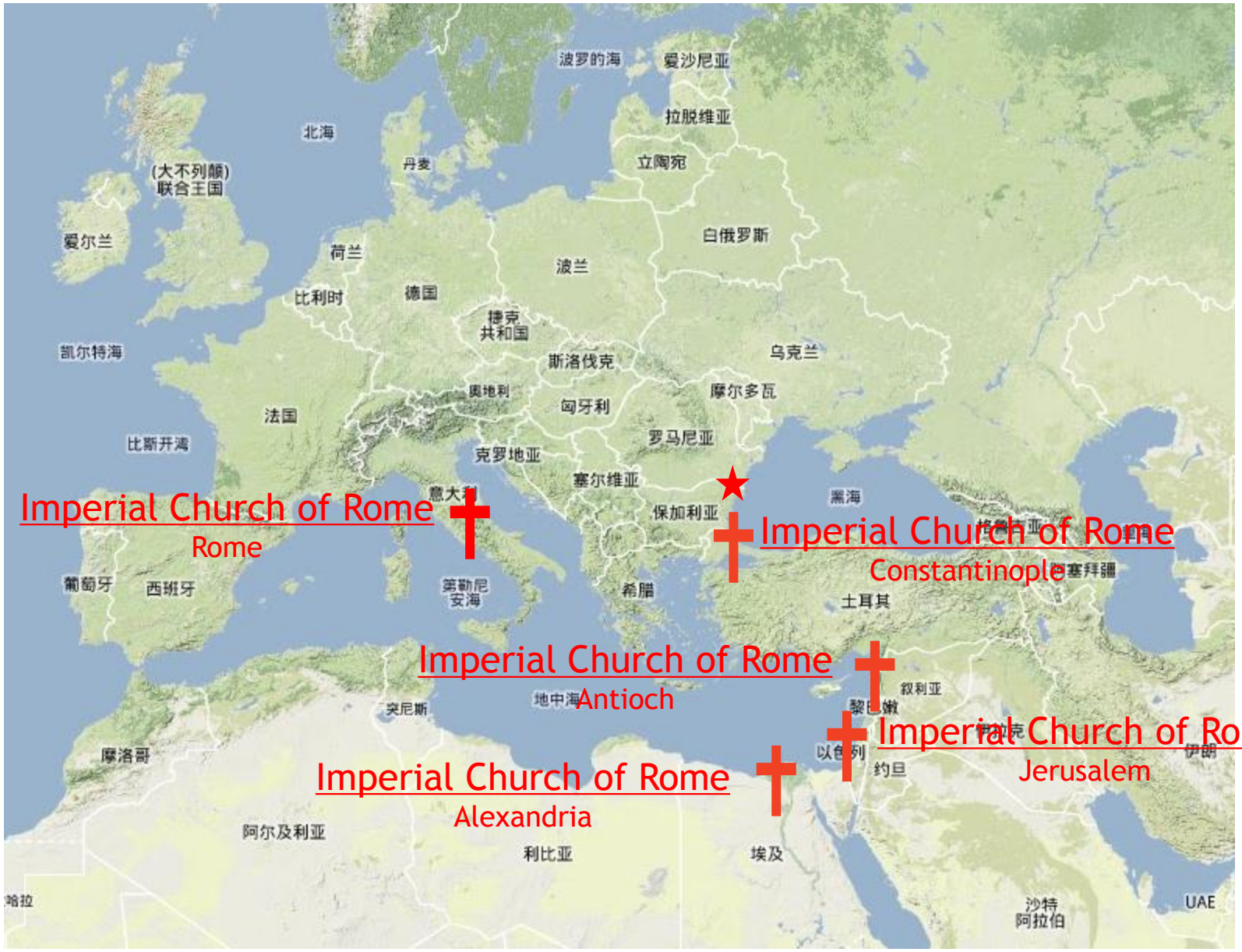
* Meaning of Life
  * To see the plan of God and to appreciate every thing that God has created for us
  * To understand why we have made all the decisions that have in turn made every one of us unique
  * To become a better soul

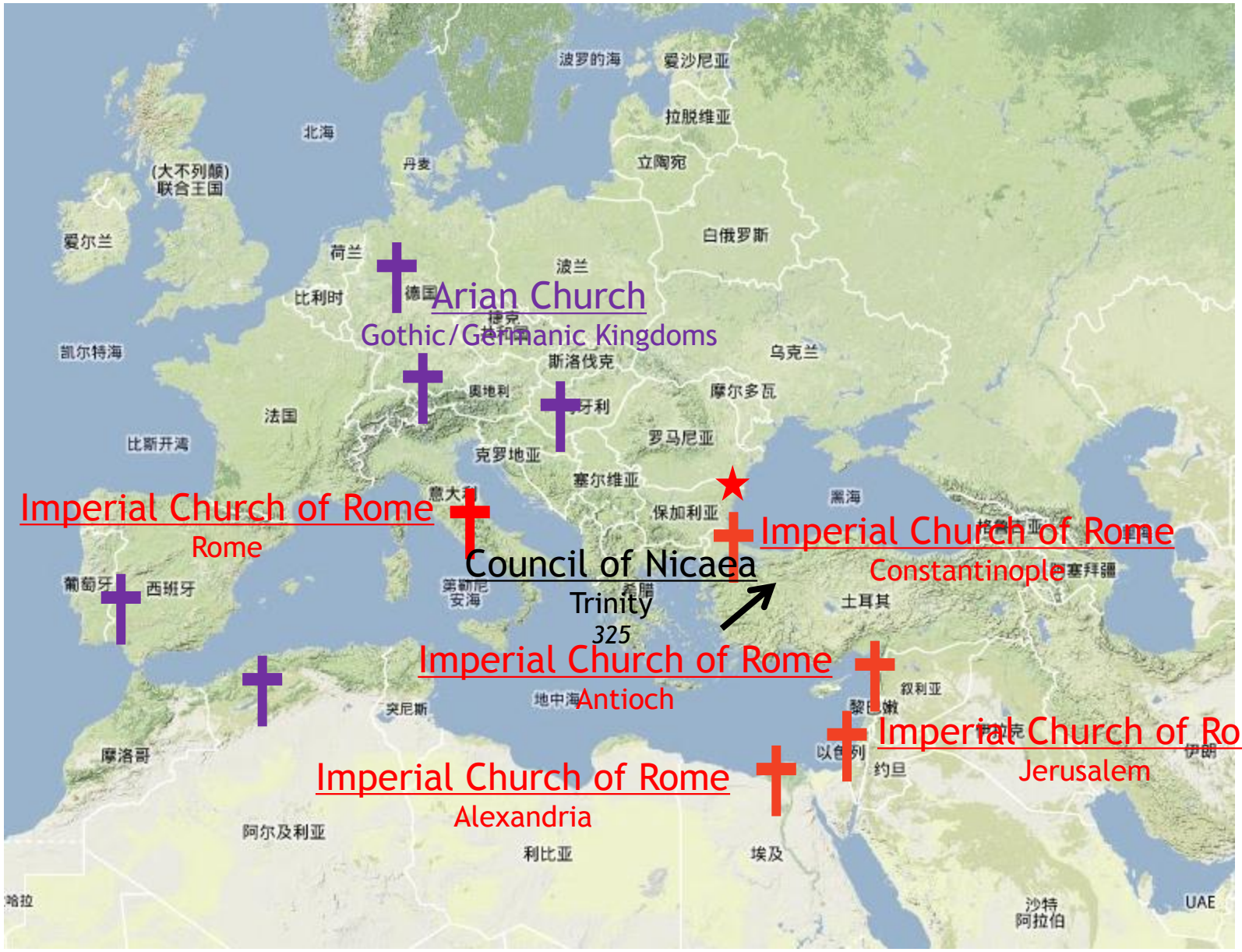
* View of the World

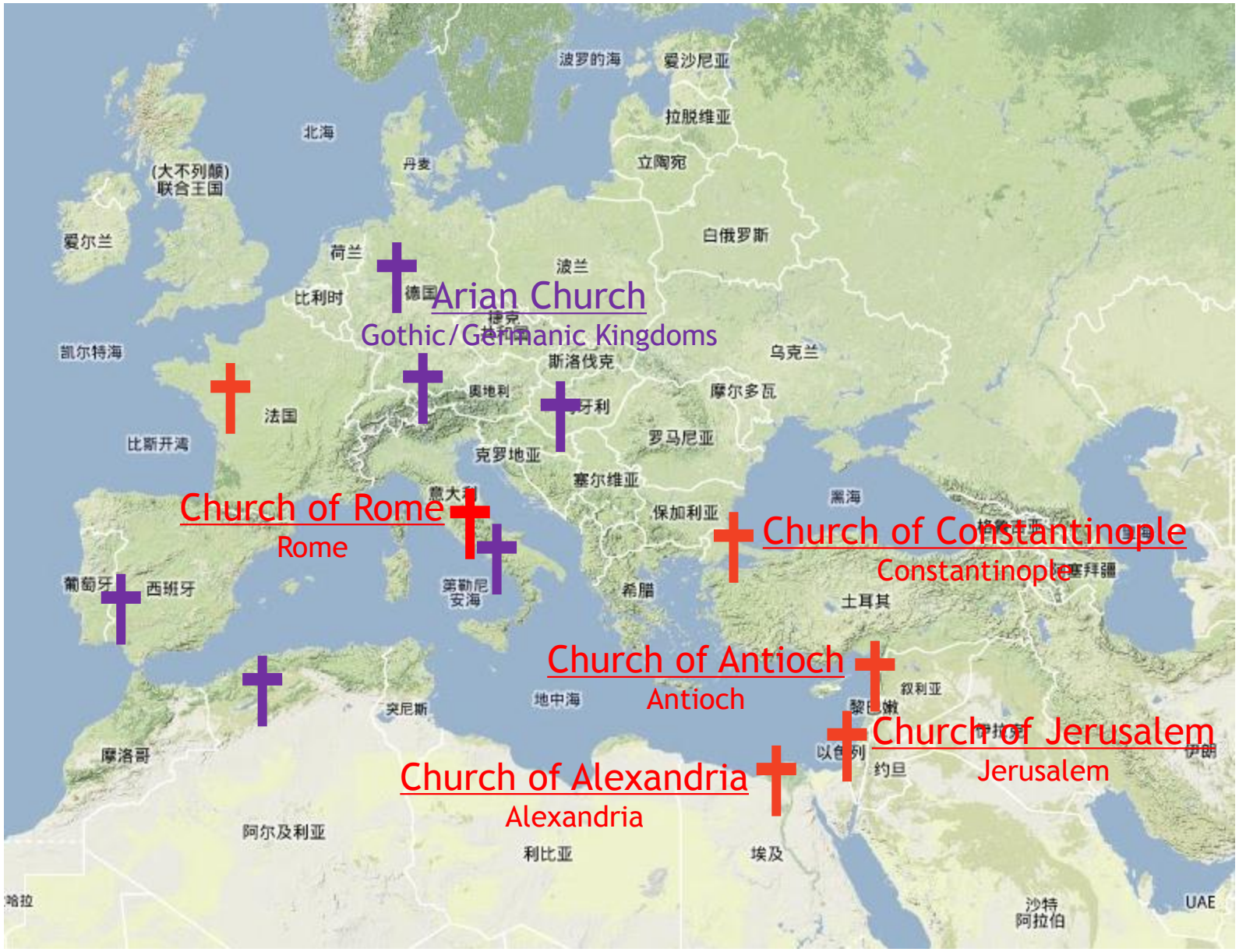
*History of Christianity

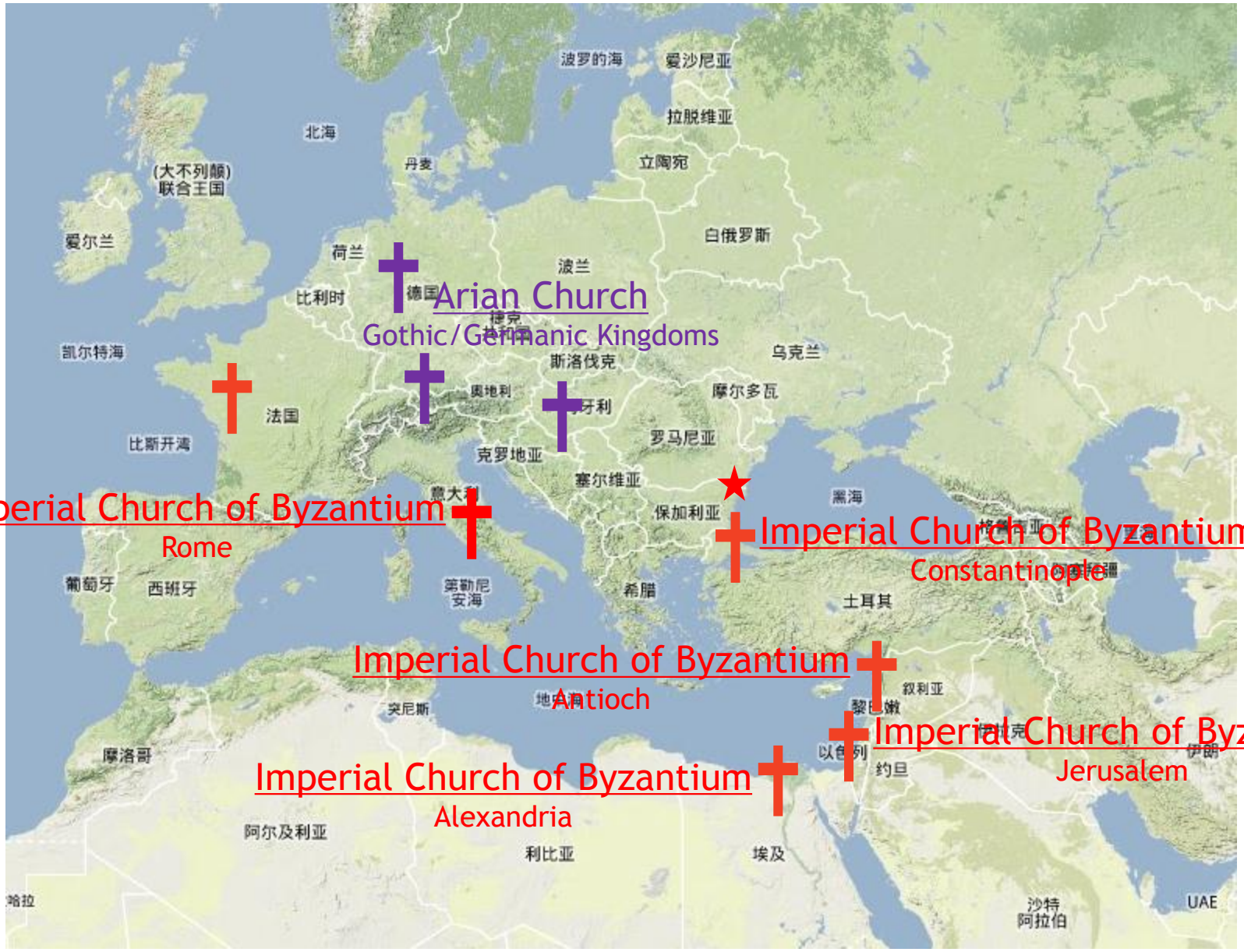

波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
白俄罗斯
荷兰
波兰
比利时
德国
捷克
共和国
凯尔特海
乌克兰
斯洛伐克
摩尔多瓦
奥地利
匈牙利
法国
比斯开湾
罗马尼亚
克罗地亚
塞尔维亚
意大利
黑海
保加利亚
格鲁吉亚
里海
阿塞拜疆
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
叙利亚
黎巴嫩
突尼斯
地中海
伊拉克
摩洛哥
以色列
约旦
伊朗
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE
Early Christianity
Jerusalem

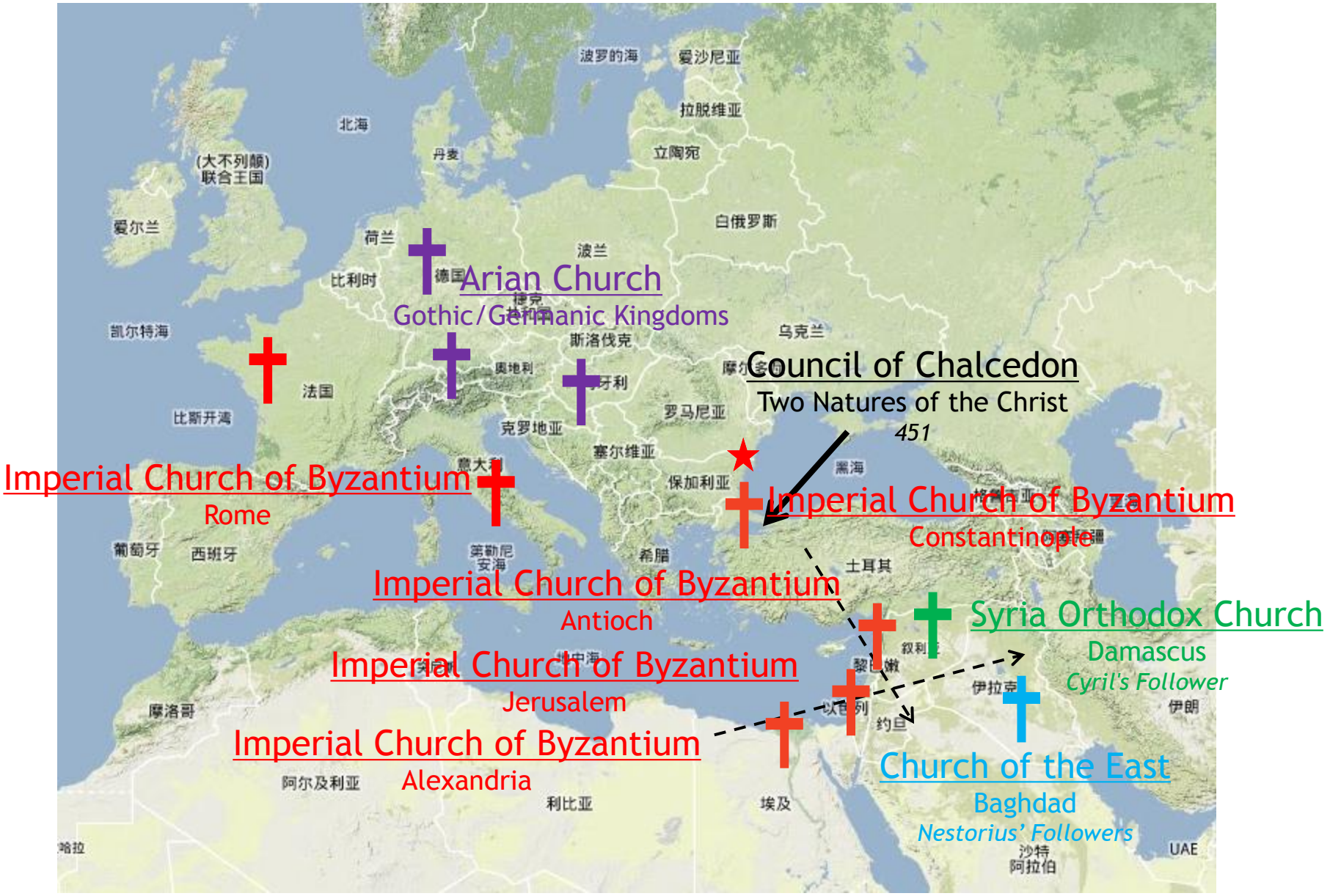

Early Christianity
Rome
Early Christianity
Antioch
Church of Edessa
Kingdom of Edessa
Early Christianity
Alexandria
波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
白俄罗斯
荷兰
波兰
比利时
德国
捷克
共和国
凯尔特海
乌克兰
斯洛伐克
奥地利
摩尔多瓦
匈牙利
法国
比斯开湾
罗马尼亚
克罗地亚
塞尔维亚
黑海
保加利亚
格鲁吉亚
里海
葡萄牙
西班牙
第勒尼
安海
希腊
阿塞拜疆
土耳其
地中海
突尼斯
黎巴嫩
伊拉克
摩洛哥
以色列
约旦
伊朗
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE

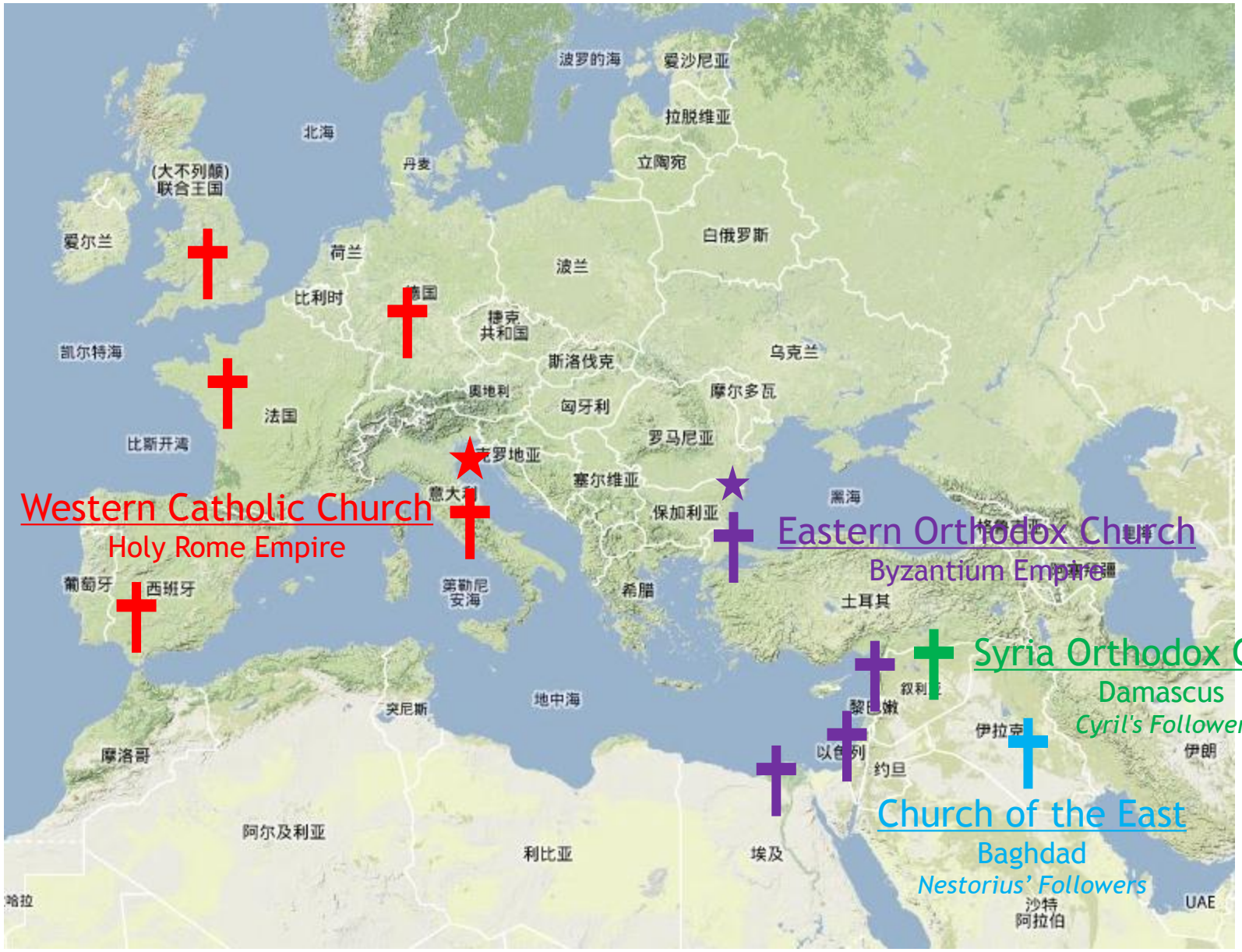

Imperial Church of Rome
Rome
Imperial Church of Rome
Constantinople
Imperial Church of Rome
Antioch
Imperial Church of Rome
Jerusalem
Imperial Church of Rome
Alexandria
波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
白俄罗斯
荷兰
波兰
比利时
德国
捷克
共和国
凯尔特海
斯洛伐克
乌克兰
奥地利
摩尔多瓦
匈牙利
法国
比斯开湾
罗马尼亚
克罗地亚
塞尔维亚
意大利
保加利亚
黑海
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
地中海
突尼斯
叙利亚
黎巴嫩
摩洛哥
以色列
约旦
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE

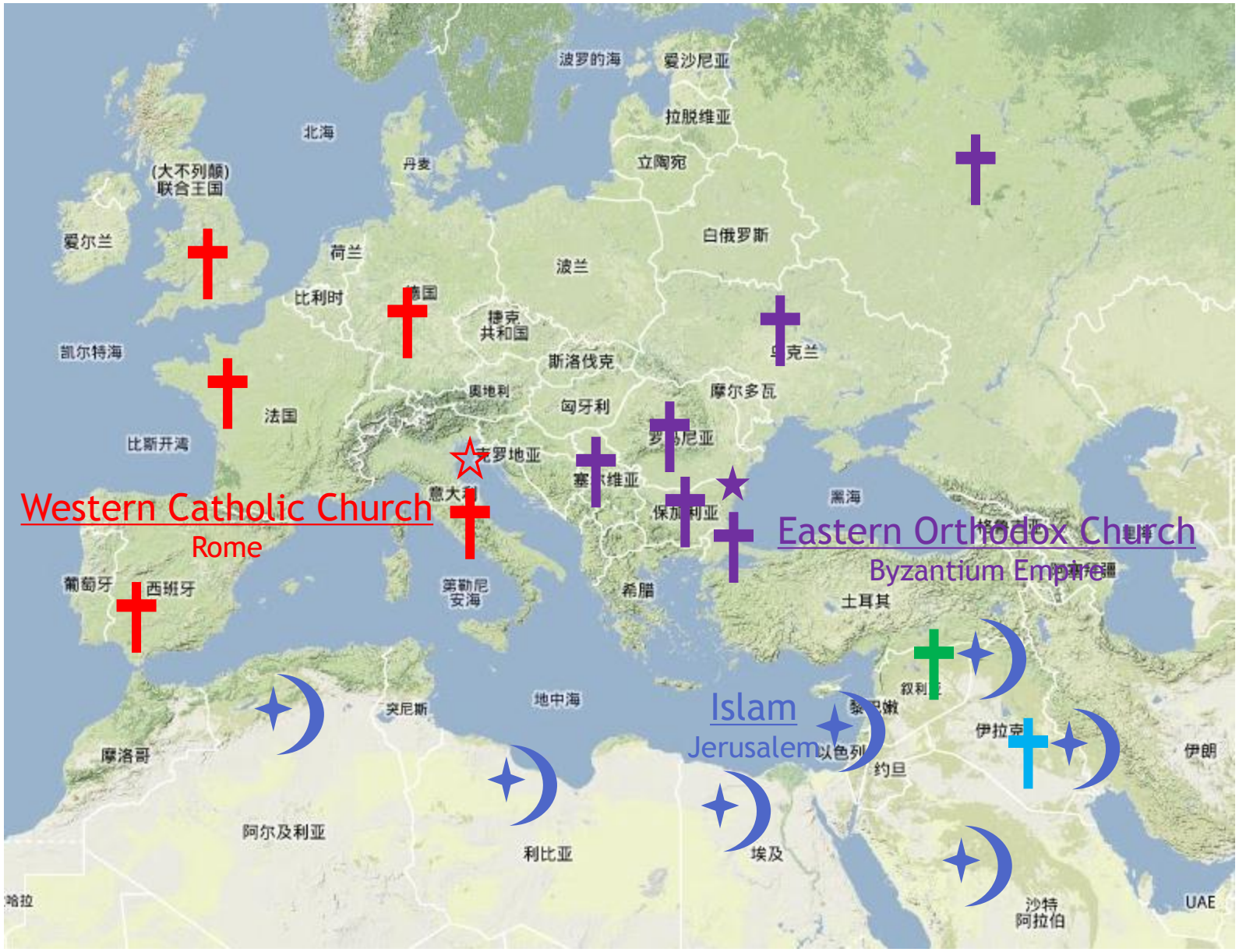

Arian Church
Gothic/Germanic Kingdoms
Imperial Church of Rome
Rome
Council of Nicaea
Trinity
325
Imperial Church of Rome
Constantinople
Imperial Church of Rome
Antioch
Imperial Church of Rome
Jerusalem
Imperial Church of Rome
Alexandria

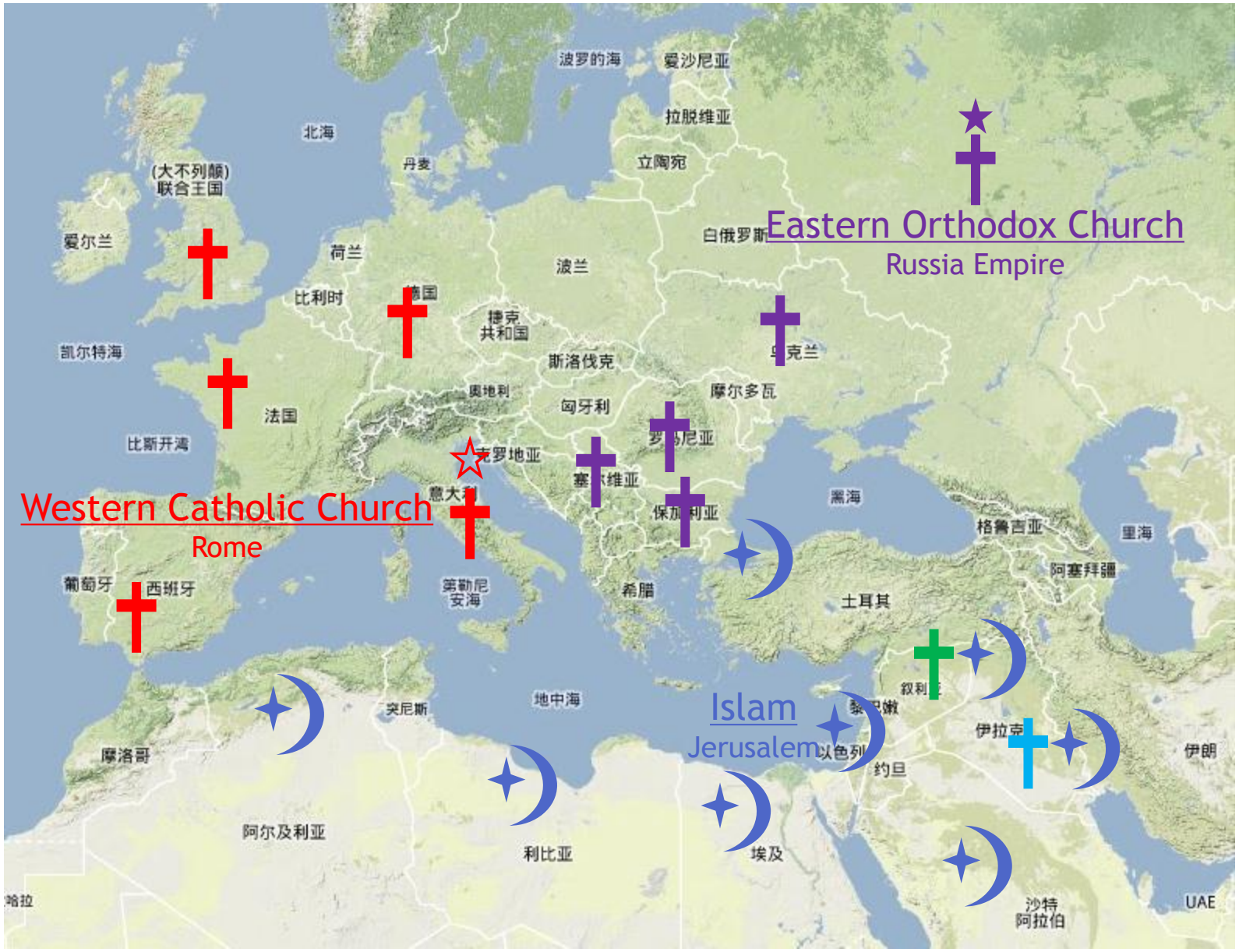

Arian Church
Gothic/Germanic Kingdoms
Church of Rome
Rome
Church of Constantinople
Constantinople
Church of Antioch
Antioch
Church of Jerusalem
Jerusalem
Church of Alexandria
Alexandria
波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
白俄罗斯
荷兰
波兰
比利时
德国
凯尔特海
斯洛伐克
乌克兰
摩尔多瓦
法国
奥地利
罗马尼亚
比斯开湾
克罗地亚
塞尔维亚
黑海
保加利亚
葡萄牙
西班牙
希腊
土耳其
地中海
突尼斯
摩洛哥
阿尔及利亚
利比亚
埃及
约旦
UAE

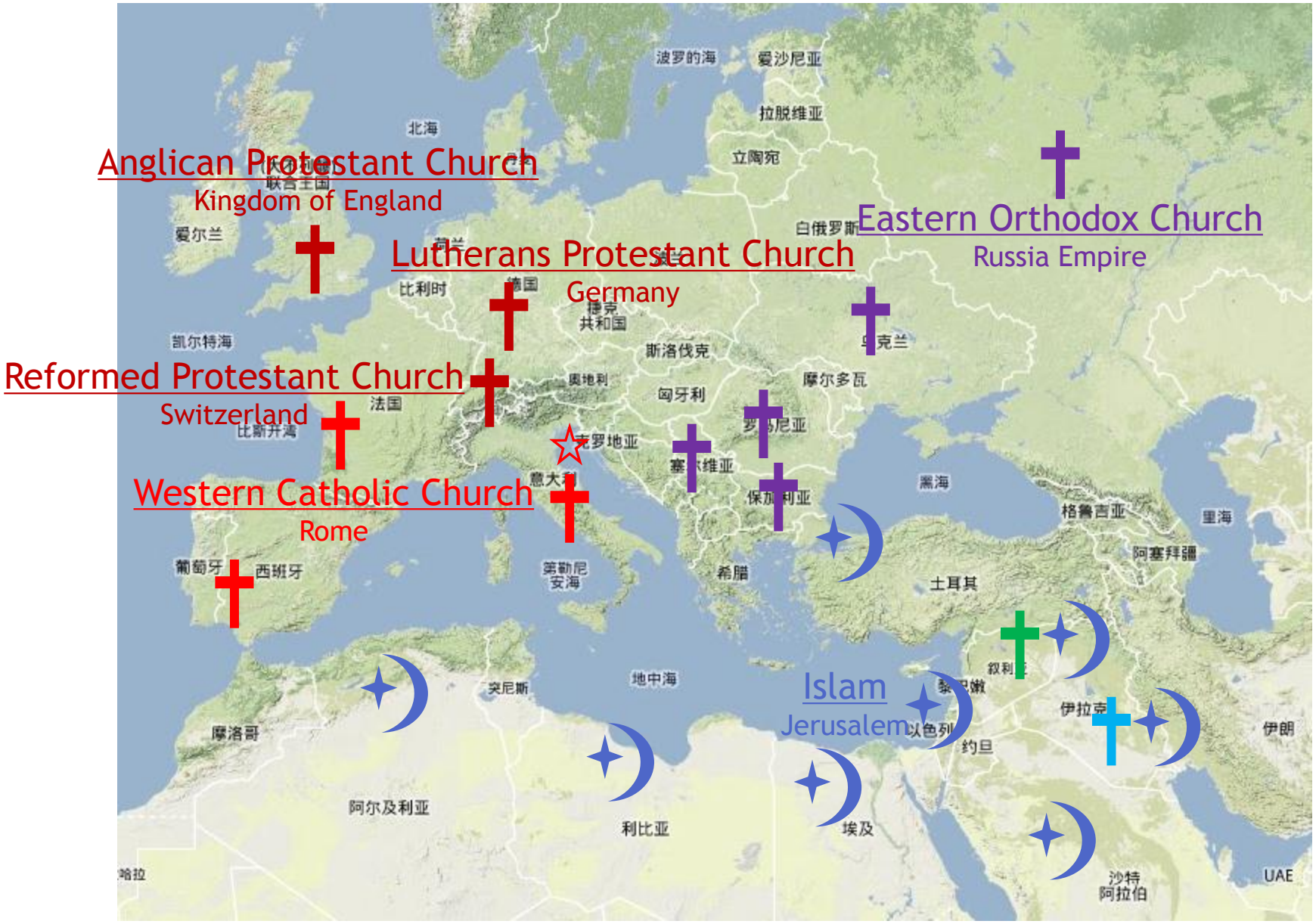

Arian Church
Gothic/Germanic Kingdoms
Imperial Church of Byzantium
Rome
Imperial Church of Byzantium
Constantinople
Imperial Church of Byzantium
Antioch
Imperial Church of Byzantium
Jerusalem
Imperial Church of Byzantium
Alexandria
波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
白俄罗斯
荷兰
波兰
比利时
德国
凯尔特海
斯洛伐克
乌克兰
摩尔多瓦
奥地利
法国
比斯开湾
罗马尼亚
克罗地亚
塞尔维亚
黑海
保加利亚
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
突尼斯
叙利亚
黎巴嫩
摩洛哥
以色列
约旦
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE


Arian Church
Gothic/Germanic Kingdoms
Council of Chalcedon
Two Natures of the Christ
451
Imperial Church of Byzantium
Rome
Imperial Church of Byzantium
Constantinople
Imperial Church of Byzantium
Antioch
Syria Orthodox Church
Damascus
Cyril's Follower
Imperial Church of Byzantium
Jerusalem
Imperial Church of Byzantium
Alexandria
Church of the East
Baghdad
Nestorius' Followers


Western Catholic Church
Holy Rome Empire
Eastern Orthodox Church
Byzantium Empire
Syria Orthodox Church
Damascus
Cyril's Follower
Church of the East
Baghdad
Nestorius' Followers
波罗的海
爱沙尼亚
拉脱维亚
北海
立陶宛
丹麦
(大不列颠)
联合王国
爱尔兰
荷兰
白俄罗斯
波兰
比利时
捷克
共和国
凯尔特海
斯洛伐克
乌克兰
奥地利
匈牙利
摩尔多瓦
法国
比斯开湾
罗马尼亚
塞尔维亚
意大利
保加利亚
黑海
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
突尼斯
地中海
摩洛哥
黎巴嫩
伊拉克
约旦
伊朗
阿尔及利亚
利比亚
埃及
UAE
沙特
阿拉伯


Western Catholic Church
Rome
Eastern Orthodox Church
Byzantium Empire
Islam
Jerusalem
波罗的海
爱沙尼亚
拉脱维亚
北海
丹麦
立陶宛
(大不列颠)
联合王国
爱尔兰
荷兰
白俄罗斯
波兰
比利时
德国
捷克
共和国
凯尔特海
斯洛伐克
乌克兰
奥地利
匈牙利
摩尔多瓦
法国
比斯开湾
克罗地亚
意大利
黑海
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
叙利亚
突尼斯
地中海
摩洛哥
伊拉克
伊朗
以色列
约旦
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE


Eastern Orthodox Church
Russia Empire
Western Catholic Church
Rome
Islam
Jerusalem
波罗的海
爱沙尼亚
拉脱维亚
北海
丹麦
立陶宛
(大不列颠)
联合王国
爱尔兰
荷兰
白俄罗斯
波兰
比利时
德国
捷克
共和国
凯尔特海
斯洛伐克
乌克兰
奥地利
匈牙利
摩尔多瓦
法国
比斯开湾
克罗地亚
意大利
黑海
格鲁吉亚
里海
葡萄牙
西班牙
第勒尼
安海
希腊
土耳其
阿塞拜疆
突尼斯
地中海
摩洛哥
伊拉克
伊朗
约旦
阿尔及利亚
利比亚
埃及
沙特
阿拉伯
UAE


Anglican Protestant Church
Kingdom of England
Lutherans Protestant Church
Germany
Eastern Orthodox Church
Russia Empire
Reformed Protestant Church
Switzerland
Western Catholic Church
Rome
Islam
Jerusalem
波罗的海
爱沙尼亚
拉脱维亚
立陶宛
北海
爱尔兰
比利时
凯尔特海
斯洛伐克
摩尔多瓦
匈牙利
奥地利
法国
比斯开湾
克罗地亚
意大利
塞尔维亚
保加利亚
黑海
格鲁吉亚
里海
阿塞拜疆
葡萄牙
西班牙
第勒尼安海
希腊
土耳其
地中海
突尼斯
摩洛哥
阿尔及利亚
利比亚
埃及
叙利亚
黎巴嫩
以色列
约旦
伊拉克
伊朗
沙特阿拉伯
UAE

*Functional Programming

What's Functional Language?

- Some may assume
  - FP language is just a tool which somehow rocks on list!
  - Declaration:

```
[1,2,3,4]; 1:[2,3,4]
```

  - Functions and Operations:

```
[1,2] ++ [3,4]; [1,2,3]!!2; head [1,2]; take 2 [1,2,3]
```

* Some may assume
  * FP language is just a tool which somehow rocks on list!
  * Texas Ranges:

```
[1..4]; [1,3..7]; [1,2..]
```

  * List Comprehension:

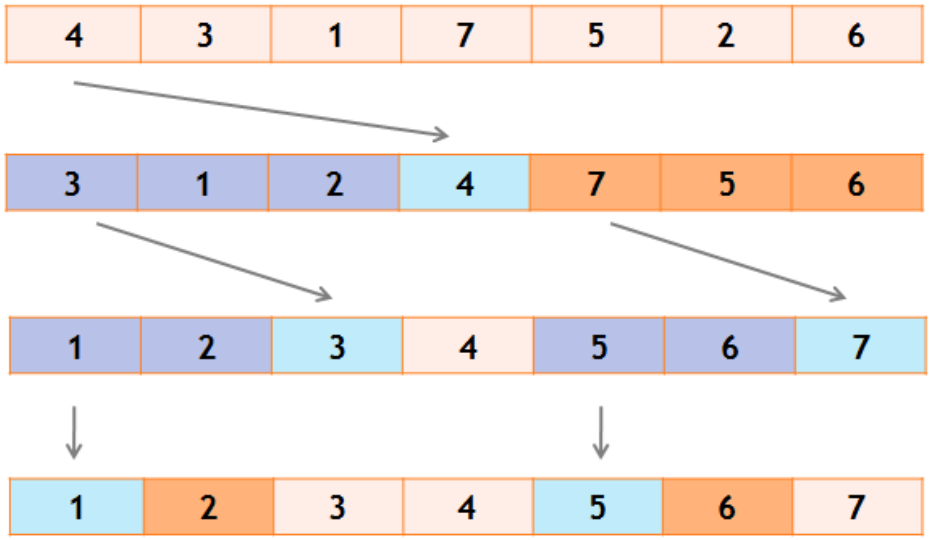
```
[x*2 | x <- [1..6], x*2 >= 12]
[(x, y) | x <- [1..3], y<- [x..4] , x*y > 6]
```

What's Functional Language?

* Imperative Programming
  * Philosophy: Program = Algorithms + Data Structure (Object-Oriented?)
  * Tuning Machine
  * C++,Java, C#, …
* Functional Programming
  * Philosophy: Program = Evaluation of Expressions
  * Lambda Expression
  * Scala, Haskell, F#, …
* Logic Programming
  * Philosophy: Program = Facts + Reasoning
  * Horn Expression
  * Prolog

*Programming Paradigm

* Quick Sort



* So What?

* C Implementation:

```
void qsort(int a[], int lo, int hi) {
    int i,j;
    ……
    qsort( a, lo, ? );
    qsort( a, ?, hi );
}
```

* Imperative Implementation

```
void qsort(int a[], int lo, int hi) {
    int h, l, p, t;
    if (lo < hi) {
        l = lo; h = hi; p = a[hi];
        do {
            while ((l < h) && (a[l] <= p)) l = l+1;
            while ((h > l) && (a[h] >= p)) h = h-1;
            if (l < h) {
                t = a[l]; a[l] = a[h]; a[h] = t;
            }
        } while (l < h);
        a[hi] = a[l]; a[l] = p;
        qsort( a, lo, l-1 );
        qsort( a, l+1, hi );
    }
}
```

*Imperative Implementation

* Haskell Implementation:

```
quicksort [] = []
quicksort (p:xs) =
    (quicksort lesser) ++ [p] ++ (quicksort greater)
    where lesser  = filter (< p)  xs
          greater = filter (>= p) xs
```

* Functional Implementation

* Imperative Language
  * HOW
* Functional Language
  * WHAT

* Cool Huh?

*Features

- Zero Side-Effects (multi-core-friendly)
- Polynomial and Overloading
- Type Inference
- High-Order Function

* Immutable variables
  * Mutable variables:

```
void foo(vector<int> &array) {
    array.push_back(1);
}
```

* No global variables
  * Global variables:

```
bool is_running;

void start() {
    is_running = true;
}
```

* Idempotent Functions!

* Zero Side-Effects

* Type System
  * class: Eq, Ord, Enum, Num, Show, …
    * Interface, [Virtual Class or Operator in C++]
  * type: Boolean, Char, Int, String, …
    * Data representation, Interface implementation, [Class in C++]
  * type variable:
    * An unknown type

* Polynomial and Overloading

* Overloading Function

```
fst :: (t1 ,t2) -> t1
fst (a, _) = a
```

* Polynomial Function

```
plus :: Num a => a -> a -> a
plus a b = a + b
```

* Polynomial and Overloading

Type Inference

- Inference
  - deduce type from expressions
- Methodology
  - type variable
  - type unification: aλ = bλ
    - a :: ([Int], Boolean), b :: ([Float], t1), λ = { Int -> Float, t1 -> Boolean }
    - aλ = ([Float], Boolean), bλ = ([Float], Boolean)

* Example

```
hello x = [x..y] where y = 2 * x
```

  * what will be the type of function hello?
    * clues: [x..y], 2*x

```
hello :: (Num a, Enum a) => a -> [a]
```

* Type Inference

Python	Weak Type	
Java	Strong Type	Type Save
C++	Strong Type	Type Unsave
Haskell	Type Inference	Type Save

*More about Type

* High-Order Function
  * a function of function
* map

```
map :: (a -> b) -> [a] -> [b]
```

* example

```
map (+3) [4,3,1,7,5,2,6]
```

* High-Order Function

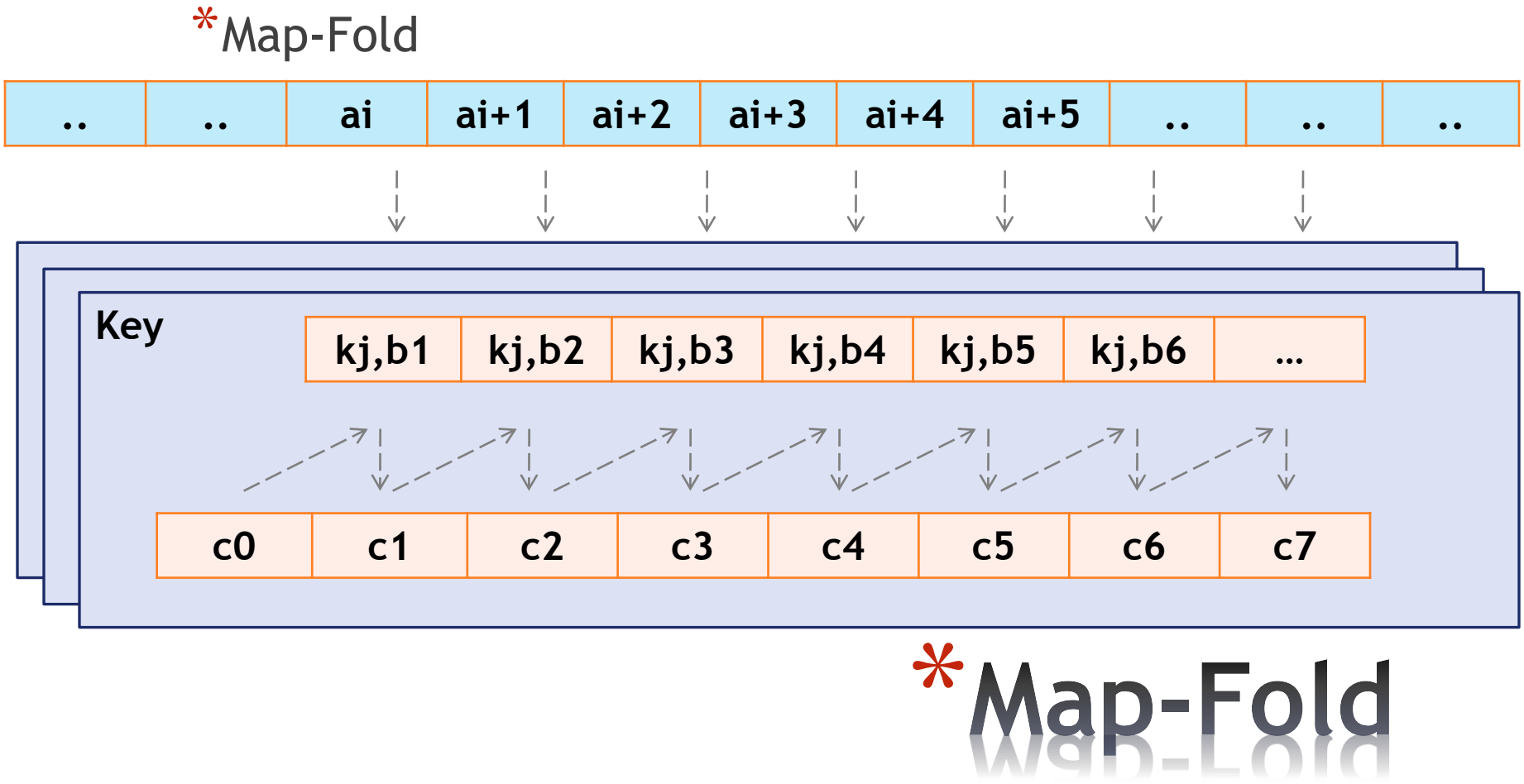
* High-Order Function
  * a function of function
* foldl

```
foldl :: (a -> b -> a) -> a -> [b] -> a
```

* example

```
puls a b = a + b
foldl plus 0 [4,3,1,7,5,2,6]
```

* High-Order Function


*Map-Fold
..
..
ai
ai+1
ai+2
ai+3
ai+4
ai+5
..
..
..
Key
kj,b1
kj,b2
kj,b3
kj,b4
kj,b5
kj,b6
...
c0
c1
c2
c3
c4
c5
c6
c7
*Map-Fold

*Thank You